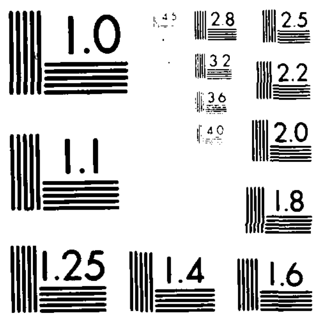
AD-A089 819 HARRIS CORP SYOSSET NY PRD ELECTRONICS DIV F/G 9/2
ATLAS TEST PROGRAM GENERATOR II (AGEN II). VOLUME I. EXECUTIVE --ETC(U)
AUG 80 J KUNG DAAB07-78-C-2015
UNCLASSIFIED CORADCOM-78-2015-F1 NL

1 OF 1
AD A
089819

END
11-80



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS

LEVEL III

(12) B.S.

RESEARCH AND DEVELOPMENT TECHNICAL REPORT
CORADCOM-78-2015-F1

A-089 796

AD A089819

ATLAS TEST PROGRAM GENERATOR II (AGEN II)
Executive Software System—Volume I

Jeffrey Kung
PRD Electronics Division
HARRIS CORPORATION
Syosset, New York 11791

August 1980

Final Report for Period April 1978—August 1980

DISTRIBUTION STATEMENT
Approved for public release;
distribution unlimited.

Prepared for:
CENTER FOR TACTICAL COMPUTER SYSTEMS

CORADCOM
US ARMY COMMUNICATIONS RESEARCH & DEVELOPMENT COMMAND
FORT MONMOUTH, NEW JERSEY 07703

80 10 1 063

NOTICES

Disclaimers

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

HISA-FM-633-78

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

19 **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 18 CORADCOM-78-2015-F1	2. GOVT ACCESSION NO.: AD-A089819
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): 6 ATLAS Test Program Generator II (AGEN II). Executive Software System, ~~Volume I~~ Volume I.	5. TYPE OF REPORT & PERIOD COVERED: 9 Final Report, Apr[redacted]78 - Aug[redacted]80
	6. ~~PERFORMING ORG. REPORT NUMBER~~
7. ~~AUTHOR(s)~~: 10 Jeffrey Kung	8. CONTRACT OR GRANT NUMBER(s): 15 DAAB07-78-C-2015
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PRD Electronics Division, Harris Corp., 6801 Jericho Turnpike, Syosset, N.Y. 11791	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: A18W9773901H-2510
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Communications R&D Command, ATTN: DRDCO-TCS-M, Fort Monmouth, N.J. 07703	12. ~~REPORT DATE~~: 11 Aug[redacted]80
	13. NUMBER OF PAGES: 12 94
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): N/A	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

User's Guide published in Volume II → The objective of the project was to design and implement

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automatic Test Program Generation
Automatic Test Equipment
Test, Measurement and Diagnostic Equipment

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the final report on the Army-CORADCOM-sponsored project which had as its objective, the design and implementation of a software processor which would automate the process of generating source code for test programs of various linear analog circuits. The approach used by AGEN (ATLAS Generator) is to take one type of linear analog circuit, e.g., amplifier, and subject it to an in-depth circuit analysis to determine the commonalities among similar types of circuits. Once the commonalities are defined, a circuit model is structured by ATLAS test language

392 13

cont

through common data base FORTRAN software techniques. The resultant ATLAS model source program is basically a standardized, high quality, error-free ATLAS test program with the variables opened as windows for the AGEN users to insert the values for their individual test specifications. Therefore, the AGEN program is an interactive test program which requires only test specifications from the user to complete the program generation.

SUMMARY

This is the final report on the Army-CORADCOM-sponsored project which, had as its objective, the design and implementation of a software processor which would automate the process of generating source code for test programs of various linear analog circuits. Previous computer-generated testing processors had been developed mainly for digital testing, and had been behind the state of the art for most analog circuits until PRD developed ATLAS Test Program Generator I (AGEN I) in 1978. The approach used by AGEN (ATLAS Generator) is to take one type of linear analog circuit, e.g., amplifier, and subject it to an in-depth circuit analysis to determine the commonalities among similar types of circuits. Once the commonalities are defined, a circuit model is structured by ATLAS test language through common data base FORTRAN software techniques. The resultant ATLAS model source program is basically a standardized, high quality, error-free ATLAS test program with the variables opened as windows for the AGEN users to insert the values for their individual test specifications. Therefore, the AGEN program is an interactive test program which requires only test specifications from the user to complete the program generation.

AGEN I was a project to demonstrate the concept on two analog circuits, i.e., amplifier and oscillator. It was tested on a PRD-designed Automatic Test Equipment (ATE)-VAST (Versatile Avionic Shop Tester) System. AGEN II is a second phase of AGEN program to demonstrate the expandability of AGEN I to other analog circuits and to other ATE's, e.g., EQUATE System. This final report will describe all these achievements of the AGEN II.

The AGEN II final report consists of three volumes:

Volume I	-	AGEN II Executive Software System
Volume II	-	AGEN II User's Guide
Volume III	-	AGEN II Software Modules Listing

Volume I, which follows this abstract, describes in detail the FORTRAN executive system which formulates the ATLAS skeleton programs and controls the flow of user-inserted variables. Volume II describes the step-by-step procedure of performing interactive ATLAS test program generation by using AGEN II and the detailed information of each analog circuit under test in terms of test theories and test flow charts. Volume III consists of the program listing for all of the software modules.

The AGEN automated analog test program source code generation concept has been proven feasible, well structured, and expandable. However, only the initial goals of an analog ATPG (Automatic Test Program Generation) have been achieved. In ATPG, modeling is the first phase, simulation by linking these models is generally the second phase, and automatic fault isolation the third phase. AGEN III, if designated for the next phase, should be an expansion of AGEN II, linking various modules to simulate a complex analog network and to fault-isolate automatically to the basic modules if failure occurs. These are the real ultimate goals of ATPG.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I.

INTRODUCTION

The AGEN II System is a computer-aided executive system used for automatic source code generation of ATLAS test programs for functional testing of certain well-defined categories of analog networks. The ATLAS test programs will be generated in a brief interactive session between the user and the AGEN II Software Program. The questions asked by the AGEN II Program will be simple and the user's answers will generally be obtainable from the manufacturer's specification sheet for the Unit Under Test (UUT). The user of the system will not be required to write network equations or possess any knowledge of computer programming techniques in order to operate the system. Furthermore, the user input is purely numerical engineering test data, which obviates the need for expertise in any specific ATE programming language.

The AGEN II System is implemented in FORTRAN IV on a UNIVAC 1108 under the EXEC VIII Operating System. An overview of the AGEN II System is given in Section II, and a detailed description of its modules is presented in Section III. AGEN II is an on-line, interactive, step-executable system, and is invoked on demand mode through a terminal or a CRT. The output of AGEN II is a magnetic tape containing ATLAS source test programs for the UUT's. This source magnetic tape is then loaded on the EQUATE Test Station and compiled by the ATLAS compiler. Thereafter, testing of the UUT's may proceed. The organization of ATLAS Test Program Library, the EQUATE Test Station, and their relationship to AGEN II as an integrated test system is presented in Sections IV and VI. The step-by-step operation of AGEN II is documented in the AGEN II User's Guide.

The AGEN II System is an extension of the original AGEN I version which only addresses amplifiers and oscillators. The current version of AGEN II accommodates three (3) additional analog networks and each network features multi-characteristics. The three (3) analog networks covered by AGEN II are filters, mixers, and power supplies. The extension of AGEN II demonstrates the capability of AGEN to expand horizontally (adding more analog networks to AGEN) and vertically (adding more characteristics to an analog network). A more detailed discussion of expandability of AGEN II is provided in Section V.

The purpose of this report is:

A. To provide a conceptual understanding of the AGEN II System.

B. To describe design and implementation features.

C. To provide detailed descriptions of each of the system components and modules and their corresponding flowcharts.

D. To describe methods of expanding AGEN II System and adding test modules into the ATLAS Test Program Library.

E. To describe the inter-relationships among the AGEN II System, ATLAS Test Program Library, and EQUATE Test Station.

For more information concerning AGEN II, refer to the following manuals:

1. AGEN II User's Guide
2. AGEN II Software Modules Listing

SECTION II.

AGEN II SYSTEM OVERVIEW

2.1 DESIGN FEATURES

The AGEN II System is designed to the following guidelines:

A. Adapt a top-down design for systematic implementation and integration. Isolate functional components by modular construction for easy modification and future expansion.

B. Interface with the user in an on-line conversational mode. Guide the user interactively by using basic engineering language so that minimum technical information is required, and operator training is minimized.

C. Limit data entries to mainly numerical, with minimal usage of the English alphabet, e.g., "y" for yes and "n" for no, and to be entered through a standard ASCII keyboard.

D. Display and verify limits for all parameters as they are entered. Flag corresponding error messages when errors are detected.

E. Tabulate and display all captured parameter values and units, for review and edit. Provide correction capability for all captured parameters.

F. Write all ATLAS test programs onto a magnetic tape compatible and readable by Data General NOVA/ECLIPSE mini-computer which controls the EQUATE Test Station.

2.2 AGEN II SOFTWARE SYSTEM ORGANIZATION

The organization of AGEN II Software System is based on functional levels with major subdivisions as AGEN, DNSC, IVT, OTPT, and OUTAPE. The software routines comprising the AGEN II Software System are constructed in modular format. Information flows between the functional levels and the modules by means of a common data base. Figure 2.2-1 shows the AGEN II Software Family Tree, and Figure 2.2-2 shows the AGEN II Functional Flowchart.

Each software level has a well-defined function to perform. As depicted in Figures 2.2-1 and 2.2-2, the AGEN module is at the top level. It serves as the executive program to direct the logical execution of relevant software modules to accomplish the task of generating the required ATLAS test program for the desired UUT. The DNSC module is the second level serving as the UUT-determiner to differentiate the networks and characteristics based on the N code and C code input from the user. The third level is the IVT module. The IVT consists of a group of five (5) software submodules. These five (5) IVT submodules are named after the N code (refer to the N Table in Section 3.2) of the networks. IVT1, for example, is the amplifier network which is represented by N=1. Each of these five (5) IVT submodules serves as the parameter-collector for its corresponding network. They collect all the necessary parameter values from the user and calculate the expected output values for their corresponding networks and characteristics. The fourth level is the OTPT module. The OTPT also consists of a group of five (5) software submodules, and each one is organized as multi-entry subroutine. The number of entries in a subroutine depends upon the number of characteristics of the network that the subroutine belongs to. The multi-entry subroutines are named after the N code and C code of the networks and characteristics. OTPT 31 is the power supply network which is represented by N=3 and its Regulation Characteristics is represented by C=1. Each of these multi-entry subroutines serves as the ATLAS Program Writer. They retrieve the appropriate preformatted ATLAS modules from the ATLAS Library, insert the user specified parameters, calculate the expected output values, and write out the resultant ATLAS program to a temporary file for their corresponding network and characteristic. The fifth level is the OUTAPE. The OUTAPE module has the capability of converting the resultant ATLAS test programs to seven bit ASCII and to write the converted ATLAS test programs to a magnetic tape compatible and readable by the Data General NOVA 3/ECLIPSE mini-computer.

The software routines of the AGEN II system is designed modularly. The modular construction is arranged to separate functional components of the system and to facilitate implementation, modifications, and future expansion. One segment or module may be implemented, debugged, modified, or extended with minimum programmer interaction or system disruption. However, the advantages of segmentation will be lost without a rigid specification of the functions, interfaces and the flow of information in the system. From this point of view, AGEN II was designed and implemented in a top-down fashion. The integrity of informational flow between functional levels and modules is achieved by utilizing a common data base. The common storage arrays, working variables, and tables are defined by common procedures. The inclusion of this common procedure in the software modules will insure inter-module communication and reduce programming redundancy.

The AGEN II software system is formatted to insure system integrity and expandability. The expansion of AGEN II from AGEN I has itself demonstrated these properties of the system.

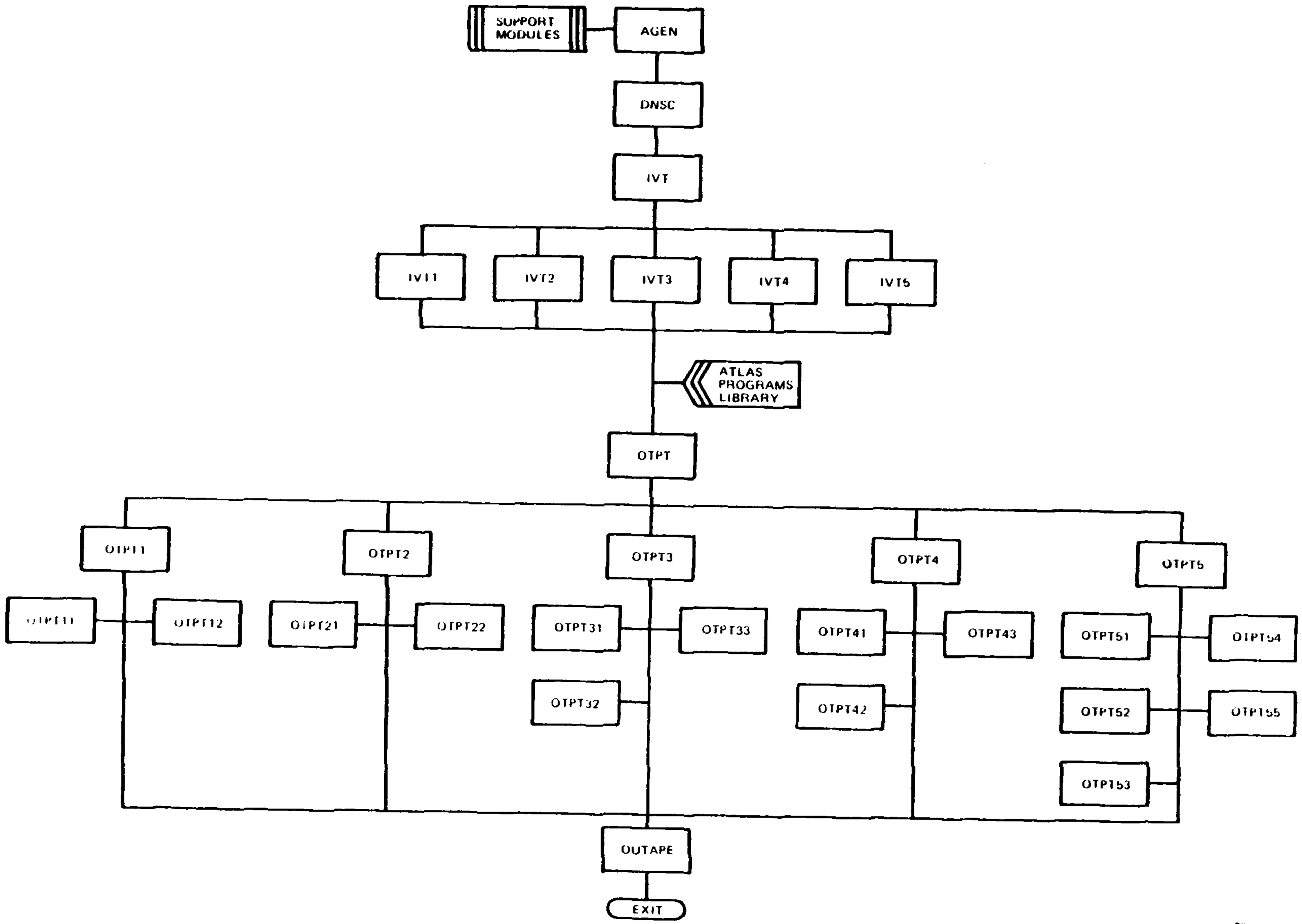


Figure 2.2-1. AGEN II Executive Software System Family Tree

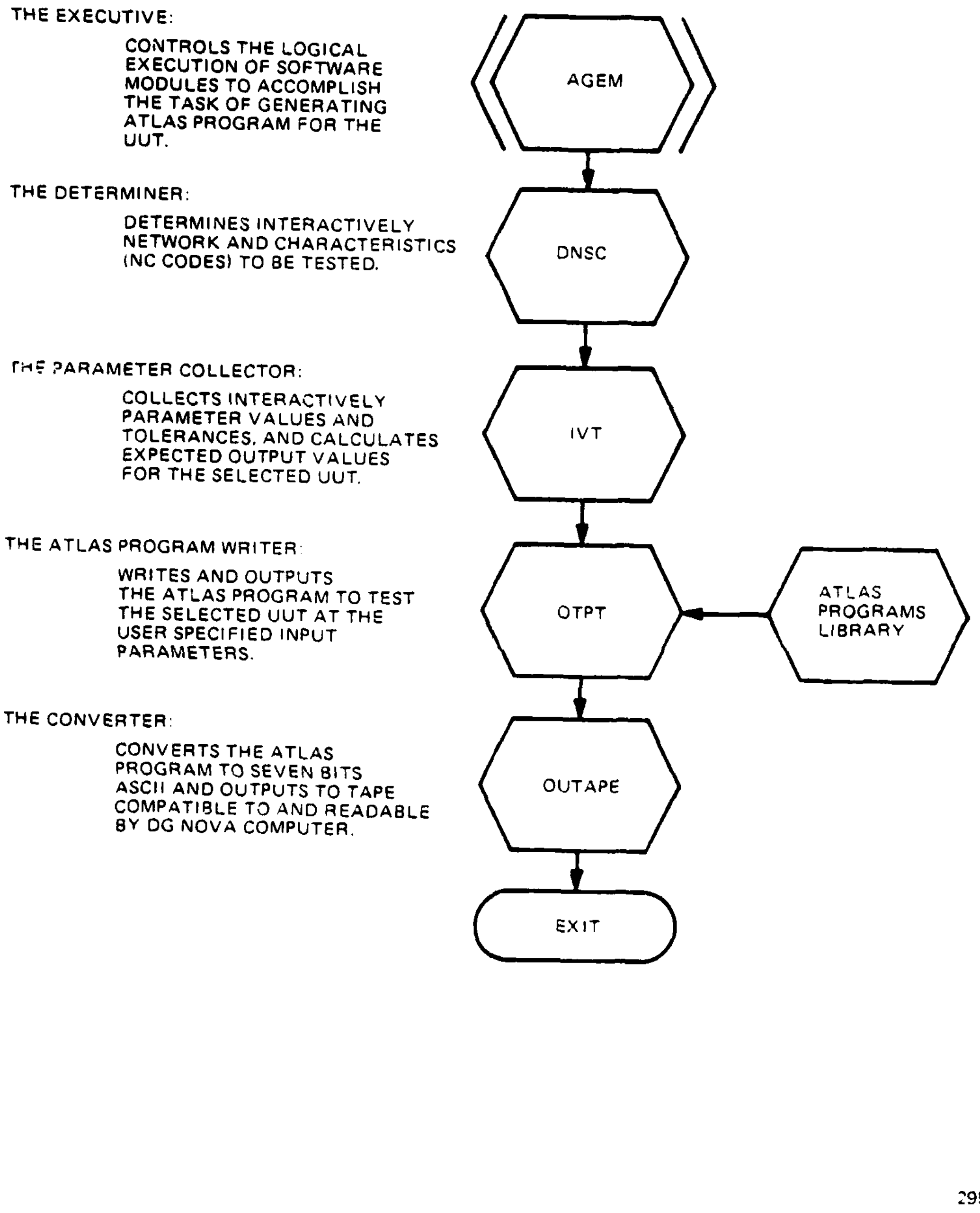


Figure 2.2-2. AGEN II Functional Flowchart

SECTION III

AGEN II SOFTWARE MODULES

The following software modules constitute the AGEN II Executive Software System. They are to be described in detail by their usage, function, organization, and flow chart.

3.1 MODULE NAME: AGEN (ATLAS GENERATOR)

3.1.1 DESCRIPTION: AGEN is the main executive program of AGEN II System. It performs the executive functions by controlling the logical execution of all necessary subroutines in order to automatically generate ATLAS programs for functional testing of Analog networks. As shown in Figure 3.1-1, the control parameters(N code and C code)needed by AGEN are provided by the DNSC module. After directing the DNSC module to obtain the appropriate N and C codes for the network and characteristics to be tested, AGEN will use these N and C codes to call other software modules to perform the necessary tasks to generate the desired ATLAS test program.

3.1.2 SUBROUTINES REQUIRED: DNSC, IVT, OTPT, OUTAPE, ETIME, TNTO, WFLE, YNCHK, OPEN, READ, WRIT, ITOH

3.1.3 PROCEDURES INCLUDED: ABAS, BBAS

3.1.4 INPUTS: User responses from terminal

3.1.5 OUTPUTS: Interactive questions, informative messages.

3.1.6 FLOW CHART: See figure 3.1-1

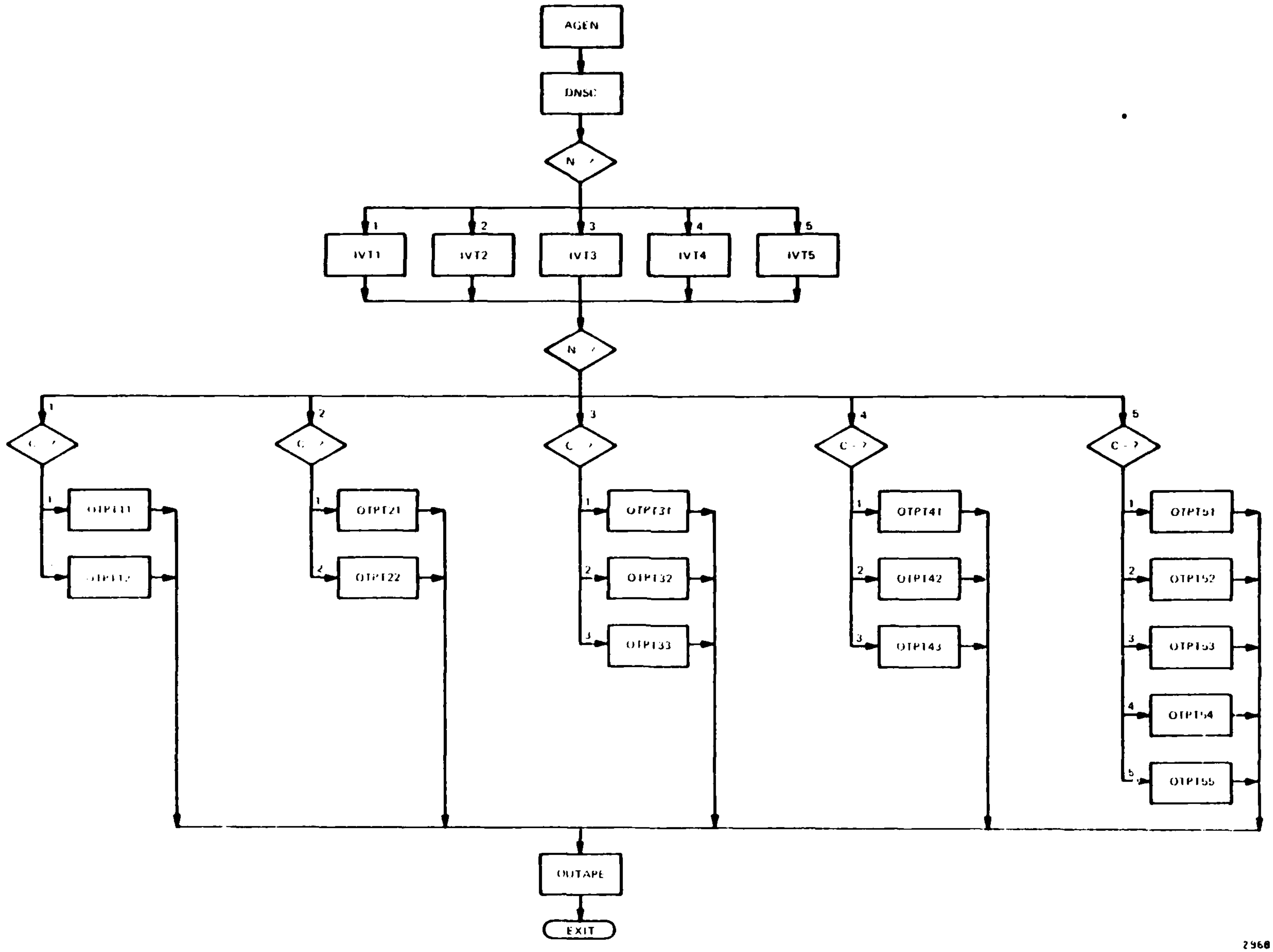


Figure 3.1-1. AGEN Control Flowchart

3.2 **MODULE NAME:** DNSC (Determine Network and Select Characteristics)

3.2.1 **DESCRIPTION:** The DNSC module determines the appropriate codes for the network (N) and the characteristic (C) codes corresponds to a unique UUT performance. The N and C codes are important control parameters for logical execution of appropriate software modules to derive the desireable ATLAS test program for the UUT. Each time the DNSC module is called by AGEN main program, the user will be able to select any one of the five (5) N codes and up to the maximum number of C codes for that network. The required codes are established via an interactive question/answer session between the user and the computer. Once the user defines the network and characteristics codes, they are fed into the computer via a terminal keyboard upon demand by the computer. In case of no knowledge of the network and characteristics coding systems, the user is guided to call for help from the computer by keying the 'HELP' or 'H' command. The 'HELP' command displays the Network (N) Table for all analog networks and characteristics (C) Table for their corresponding characteristics implemented in AGEN II.

The N and C codes are established sequentially. The first code is the N code which established the type of network to be tested. The N code Table is shown below.

NETWORK (N) CODING TABLE

N CODE	DESCRIPTION
1	Amplifiers
2	Oscillators
3	Power Supplies
4	Mixers
5	Filters

After selecting the N code, the user is requested to select the C codes of the selected N code which determines the characteristics to be tested. The user may select more than one C code and up to the maximum number of C codes for the selected network by entering 'ALL' or 'A'.

CHARACTERISTICS (C) CODING TABLE FOR AMPLIFIERS (N=1)

C CODE	DESCRIPTION
1	AMPLITUDE RESPONSE
2	LINEARITY

CHARACTERISTICS (C) CODING TABLE FOR OSCILLATORS (N=2)

C CODE	DESCRIPTION
1	FREQUENCY STABILITY
2	AMPLITUDE STABILITY

CHARACTERISTICS (C) CODING TABLE FOR POWER SUPPLIES (N=3)

C CODE	DESCRIPTION
1	REGULATION
2	RIPPLE
3	OVERCURRENT

CHARACTERISTICS (C) CODING TABLE FOR MIXERS (N=4)

C CODE	DESCRIPTION
1	ISOLATION
2	FREQUENCY RESPONSE
3	CONVERSION LOSS

CHARACTERISTICS (C) CODING TABLE FOR FILTERS (N=5)

C CODE	DESCRIPTION
1	INSERTION LOSS
2	PHASE RESPONSE
3	INPUT IMPEDANCE
4	OUTPUT IMPEDANCE
5	AMPLITUDE RESPONSE

The N and C codes are the key control parameters. AGEN II users have to enter the N and C codes for the network and characteristics to be tested and these two codes in turn control the logical sequential execution of software modules to derive the desired ATLAS Test Programs. If the user enters N=3 and C=1, for example, AGEN II will generate ATLAS Test Program for Power Supply - Regulation through a sequential execution of a set of appropriate software modules. If the user enters N=3 and C=A, on the other hand, AGEN II will generate ATLAS Test Programs for all the characteristics (Regulation, Ripple, Over Current) of Power Supplies. Therefore, a user is able to enter multiple characteristics for a given N code.

3.2.2 SUBROUTINES REQUIRED: ATOI

3.2.3 PROCEDURES INCLUDED: ABAS, BBAS

3.2.4 INPUTS: N Code and C Codes from the user

3.2.5 OUTPUTS: N Code and C Codes stored in NSC common array

3.2.5.1 NSC: NSC is a common integer-array with ten (10) elements, NSC (10), defined in ABAS. It is used to store and to communicate the N code and C codes to other software modules in AGEN II. The N code is stored in NSC (1). The C codes are stored in NSC (3) to NSC (10), and NSC (2) is not used. Pictorially, the NSC array structured as follows:

1	2	3	4	5	6	7	8	9	10
N	X	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8

NSC ARRAY

Therefore, the current NSC array can store up to maximum of eight (8) characteristics for a given network (N) which is sufficient for the current AGEN II.

3.2.6 **FLOWCHART:** See Figure 3.2-1

3.3 **MODULE NAME:** IVT (Input Values and Tolerance)

3.3.1 **DESCRIPTION:** The IVT module obtains the required input values to establish runtime variables and the acceptable tolerances of the expected outputs. The minimum number of input parameters required to generate reasonable test for the particular combination of N and C codes is solicited by this module from the user through interactive conversation mode. The user is given the facility to input parameter tolerance either in the form of absolute or variable values or in the form of percentage of nominal parameter value. By making use of the operational parameters and tolerances supplied by the user, this module calculates the upper and lower limits of the expected outputs. Since the only interface between the user and the AGEN II is through the IVT module, every input parameter and tolerance limit is checked and validated before it is accepted by AGEN II. If there is an illegal parameter value entered by the user, an appropriate error message will be displayed and the parameter value will be selected again. To further ensure the correctness of the input parameters and tolerances, a User Selected Parameters and Tolerances Table is displayed with indices at the end of parametric data entry. At this point, the user may review all the parameters and tolerances that have been entered. If there is any discrepancy on the parameter and tolerance, the user may correct the discrepancy by using the proper index which corresponds to the parameter. The correction feature covers every characteristic of each network.

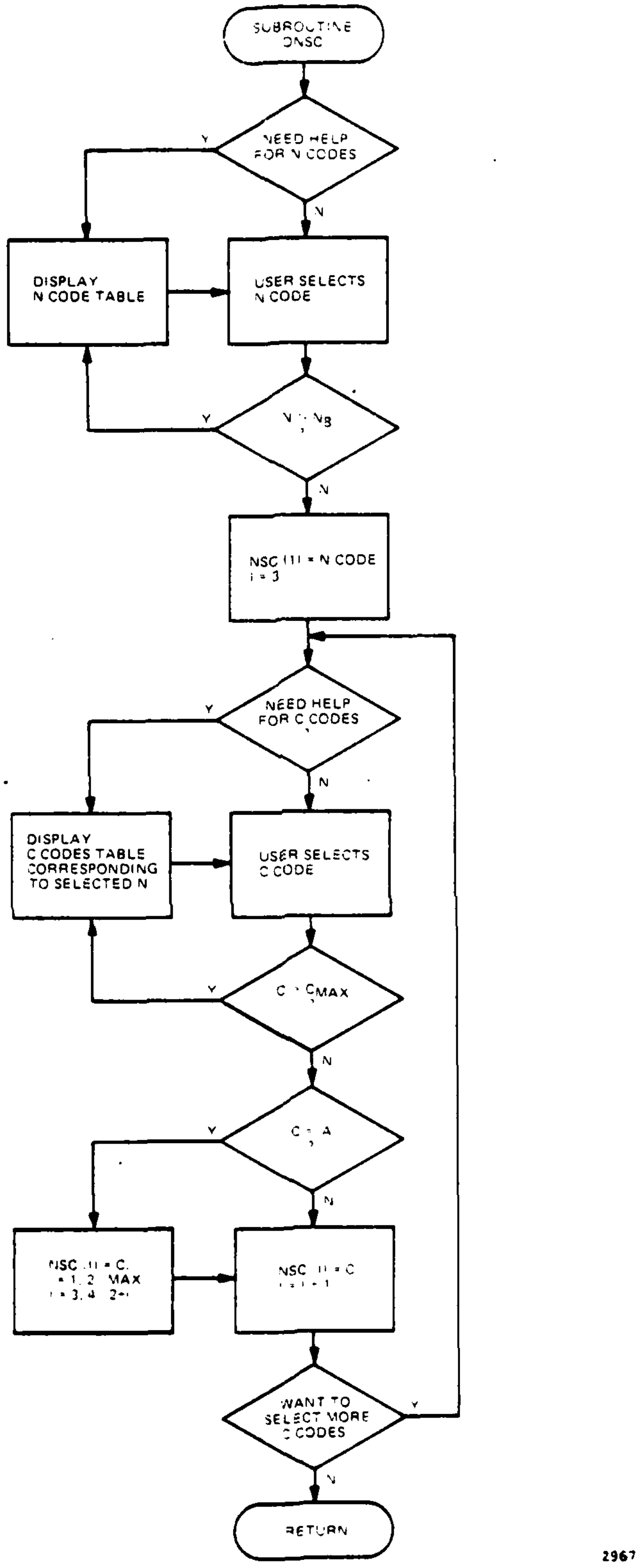


Figure 3.2-1. DNSC Program Flowchart

Additionally, the user may utilize the correction feature to change all the parameters and tolerances values entered previously. Once the User Selected Parameter and Tolerance Table is released, the process of generating ATLAS Test Programs will be carried out automatically by AGEN II. The general process of parameters/tolerances data entry, validations, corrections, and expected outputs calculations is illustrated in Figure 3.3-1.

The IVT module is expected to extract from the user the relevant, precise and minimal parametric values and tolerances of any characteristic of the five analog networks. The interactive parametric data entry module for each analog network is unique and assigned to the five (5) analog networks individually as IVT 1, IVT 2, IVT 3, IVT 4 and IVT 5. One IVT sub-module is associated to one analog network and this association is identified by the N code assigned to the network. Within each IVT sub-module, a portion is dedicated to capture the parameters and tolerances of a characteristic which is identified by the C code assigned to it. The IVT structure is illustrated by Figure 3.3-2.

The flowchart of each IVT module and the assignment of its parameters to the elements of common CARY array are attached in Appendix A.

3.3.2 SUBROUTINES REQUIRED: YNCHK

3.3.3 PROCEDURES INCLUDED: ABAS

3.3.4 INPUTS: User supply parameters and tolerances values

3.3.5 OUTPUTS: Parameters values and calculated upper and lower limits stored in CARY common array

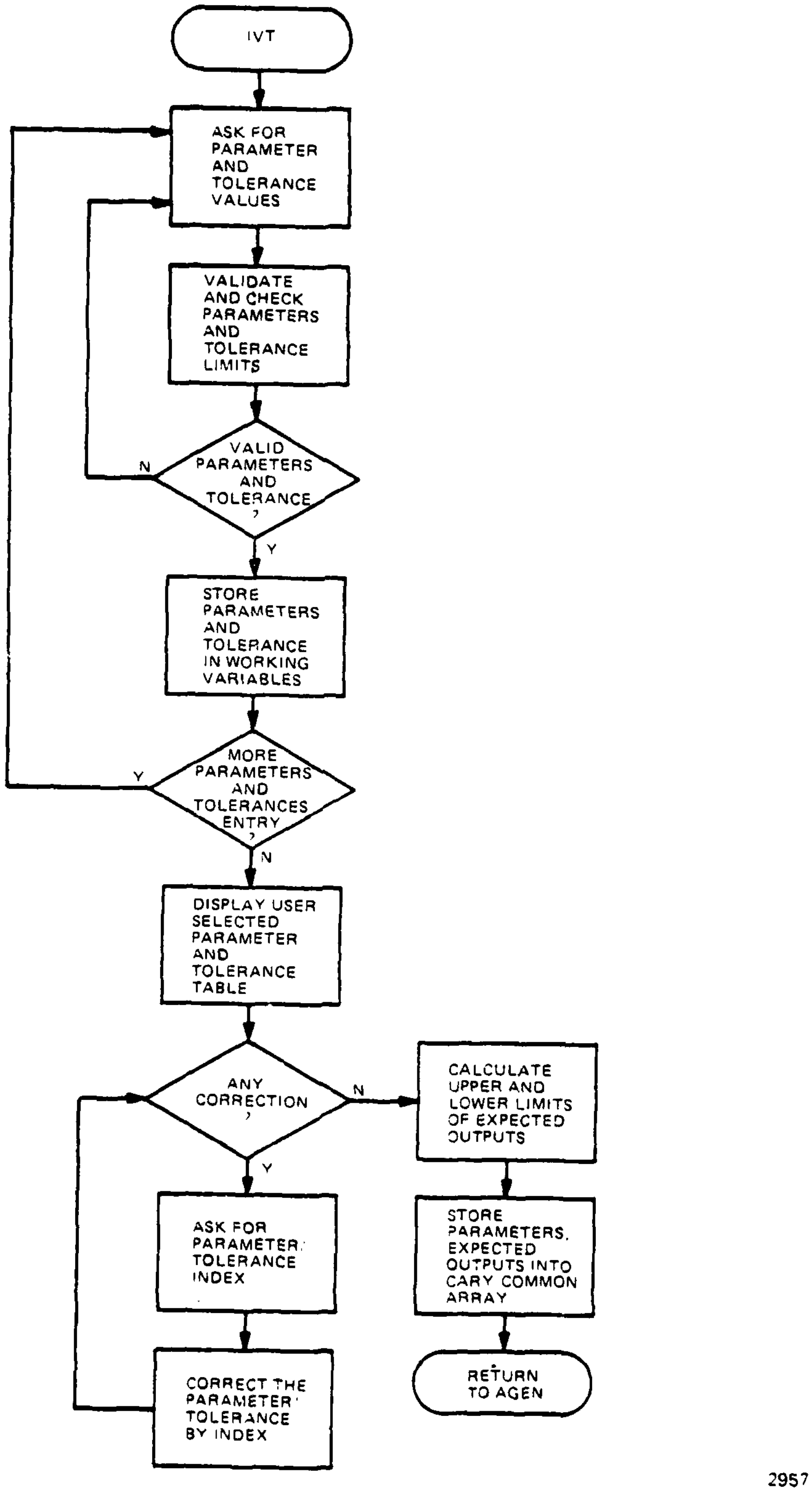


Figure 3.3-1. High-Level Flowchart of IVT Module

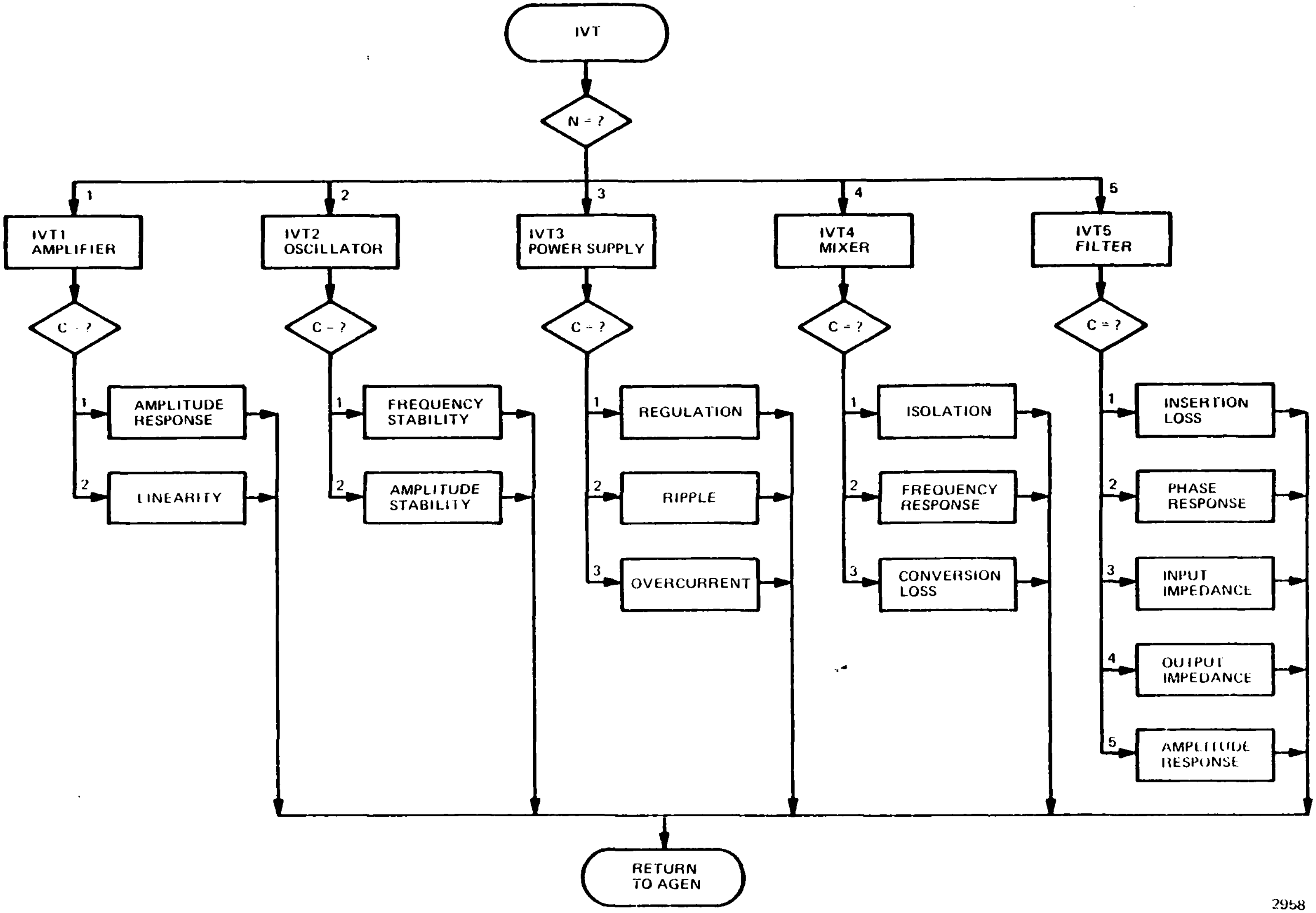


Figure 3.3-2. Conceptual Structure of IVT Modules

Table on page 6-1 of UNIVAC 1100 Series EXEC 8 Hardware/Software Summary) and insert these converted parametric data into the ATLAS runtime variable statements in appropriate format at the required places in the ATLAS test program skeleton. This task is accomplished by calling ITOH to perform the actual parametric data conversion and insertion.

E. Write the processed ATLAS statement card images to a scratch file on Drum 25. These processed ATLAS statement card images formulate the preedited ATLAS test program on the scratch file for the user requested characteristic. The final operational ATLAS program is produced after editing and copying from the scratch file to the Temporary Program File (TPFØ) by WFLE (Write File). The task of writing ATLAS statement card image to the scratch file on Drum 25 is accomplished by calling NTRAN, a UNIVAC System routine (refer to UNIVAC 1100 Series System Fortran V Library Manual), to perform the actual writing.

F. If there is any I/O error occurred during the performance of the above described I/O tasks (A to E), an I/O status code will be printed out by TERR to indicate the type of I/O error. The meaning of the I/O status code is explained on page 4-1 of UNIVAC 1100 Series EXEC 8 Hardware/Software Summary.

The supportive modules needed by OTPT to generate the operational ATLAS programs are shown in Figure 3.4-1, and the general logical process of generating the operational ATLAS programs is shown in Figure 3.4-2.

The OTPT module is expected to generate the operational ATLAS programs for any characteristic of the five (5) analog networks currently handled by AGENII. One

OTPT submodule is associated to one analog network, and this association is identified by the N code assigned to the network. Each OTPT sub-module is organized as multi-entry sub-module. The number of entries in a given OTPT sub-module depends on the number of characteristics of the analog network that this given OTPT sub-module is implemented for. The general nomenclature for the entry points of an OTPT sub-module takes the form of OTPTNC. Where N is the network code and C is the characteristic code. OTPT 12 (an entry point in OTPT 1 sub-module) which is implemented for Amplifier-Amplitude Response, for example, has N= 1 and C=2. The OTPT conceptual structure is illustrated by Figure 3.4-3.

3.4.2 SUBROUTINES REQUIRED: OPEN, READ, WRITE, ITOH, NTRAN, TNTO, TERR

3.4.3 PROCEDURES INCLUDED: ABAS

3.4.4 INPUTS: Parametric values and calculated expected outputs (provided by IVT) stored in common array CARY.

3.4.5 OUTPUTS: Pre-edited ATLAS source test programs.

3.4.6 FLOWCHART: See Figure 3.4-2.

3.5 SUPPORT MODULES

There are many support modules in the AGEN II Software System. These support modules perform many basic functions which are important for ATLAS test program generation. These support modules are described in the following sections.

3.5.1 ABAS

The ABAS is a FORTRAN Procedure where common memory areas such as arrays, working variables, tables are defined. Most of the modules in AGEN II include this procedure. The inclusion of this procedure will enable inter-modular communication because of access to the same common memory areas. A listing and explanation of usage of individual common memory area are provided in AGEN II Software Modules Listing.

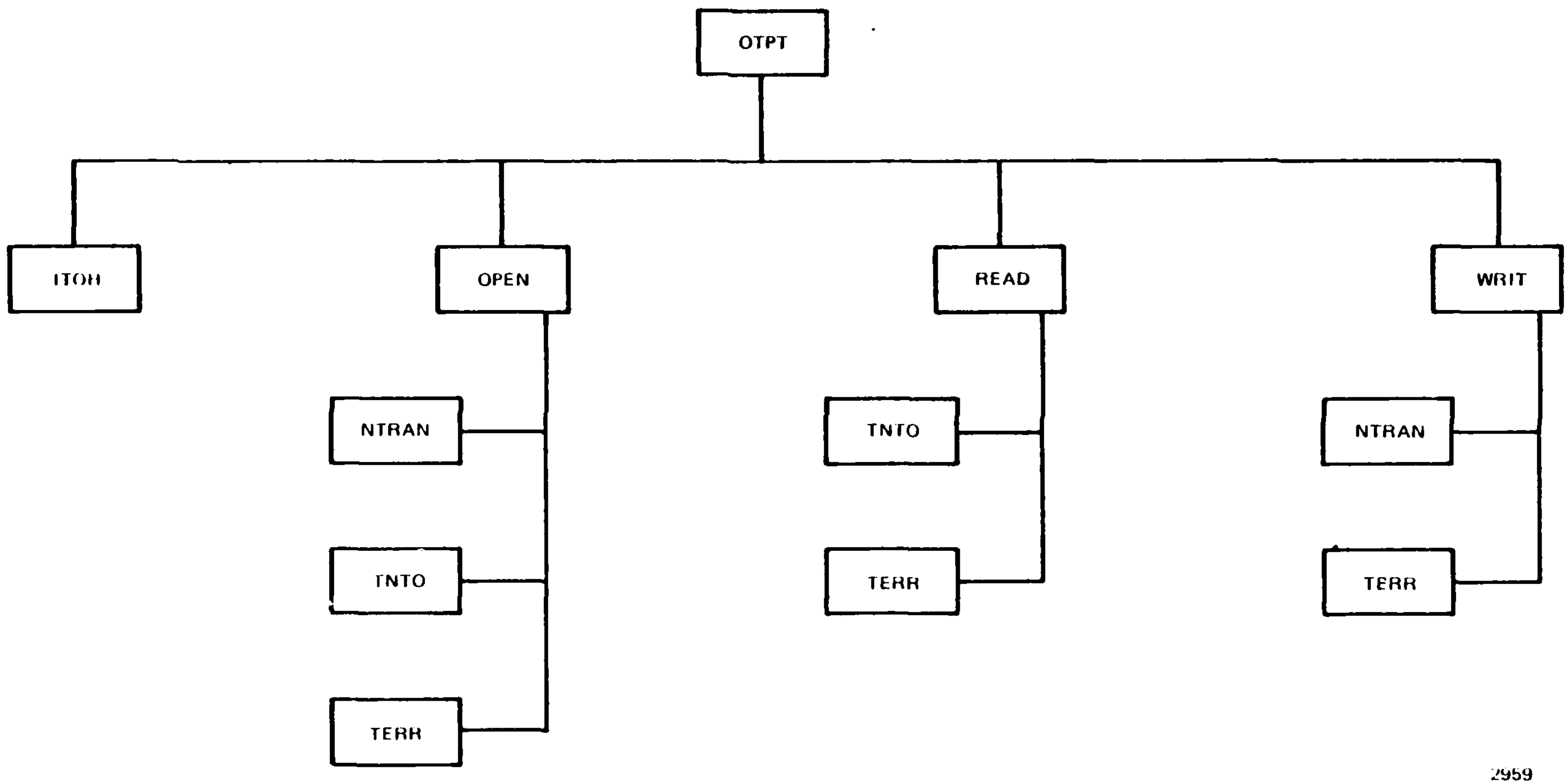


Figure 3.4-1. OTPT Support Modules Block Diagram

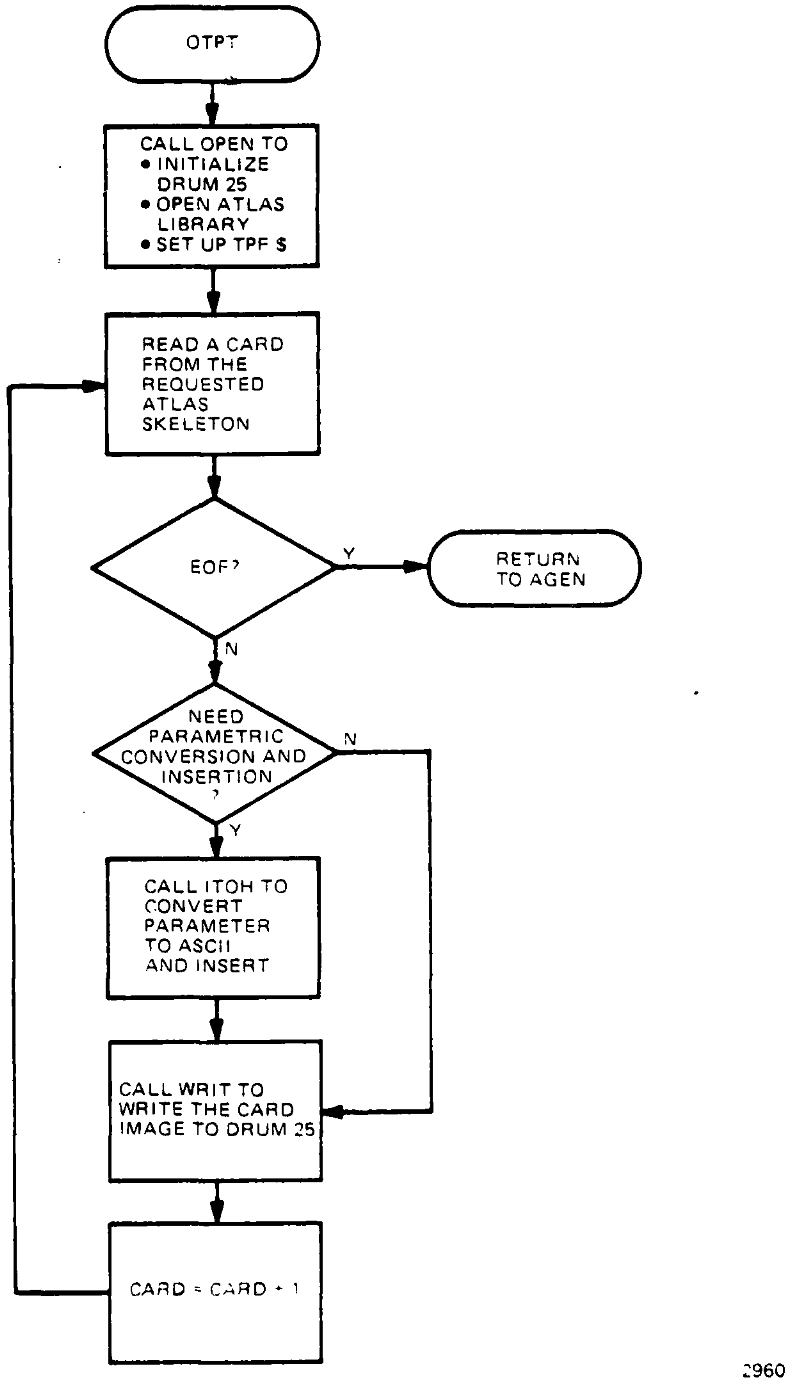


Figure 3.4-2. High Level Flowchart of OTPT Module

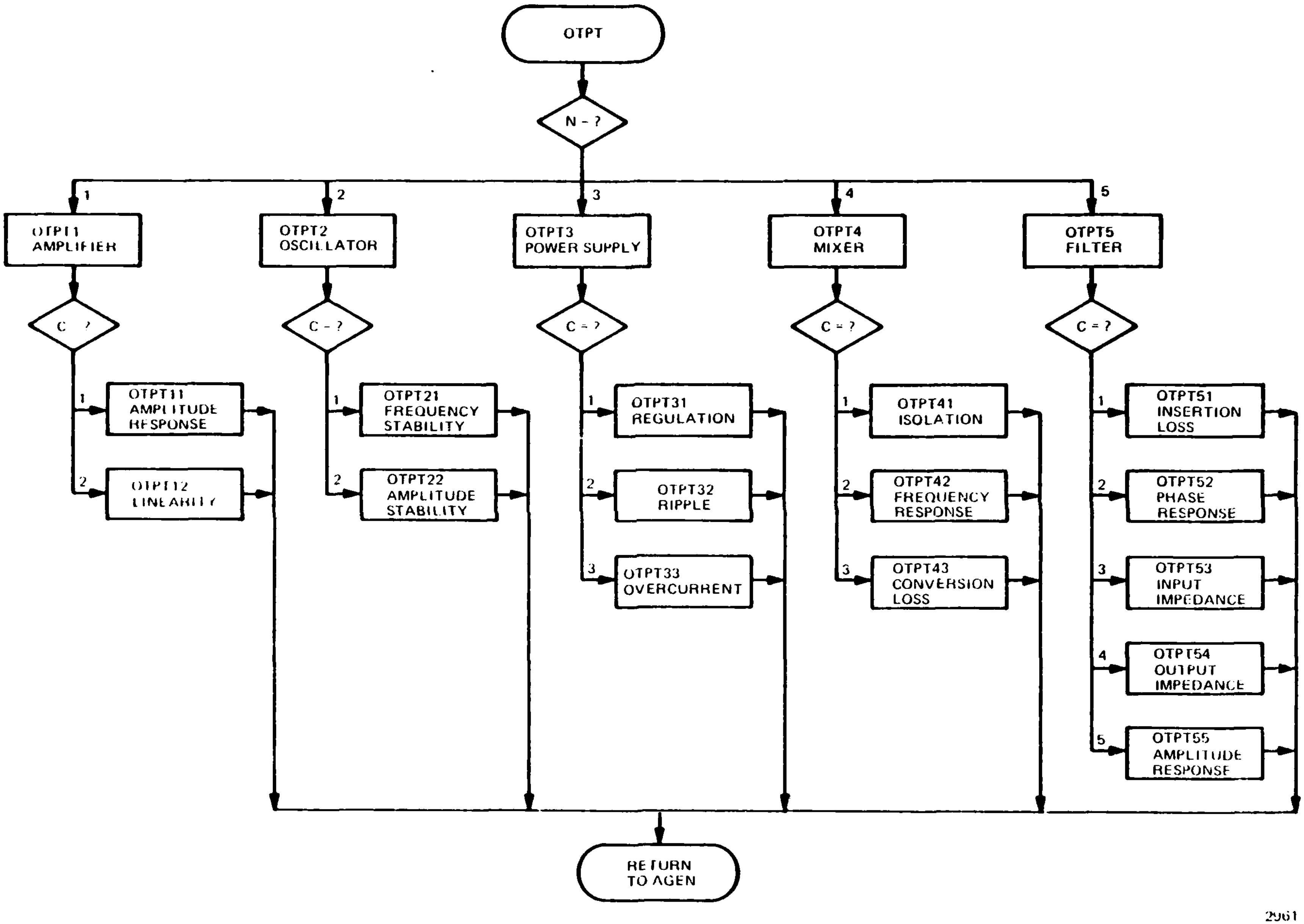


Figure 3.4-3. Conceptual Structure of OTPT Modules

3.5.2 BBAS

The BBAS is a FORTRAN procedure where the C (characteristic) code tables of the analog networks implemented in AGEN II are defined. This procedure is designed to reduce programming redundancy, to expand the C code tables easily, and to have a better software organization. Further discussion of BBAS and its expansion is found in Sections 5.3. A listing of the C code tables is provided in AGEN II Software Modules Listing.

3.5.3 ITOH

The ITOH is a FORTRAN subroutine used to convert input numeric data in field data to Octal representation of Hollerith Code. The conversion table is provided on page 6-2 of UNIVAC 1100 Series Hardware/Software Summary. The ITOH subroutine is used extensively by the OTPT modules to insert parametric values in Hollerith Code format into ATLAS test program statements. The ITOH Subroutine is defined as:

Subroutine ITOH (IN, IOUT, IDIM, KNSRT, MOCTN) where

IN: contains the input number to be converted to Hollerith Code

IOUT: contains the converted number in Hollerith Code to be outputted back to the calling program

IDIM: a variable used to dimension the I/O arrays IN, IOUT, and controls the number of times going through the conversion loop.

KNSRT: a variable used to index the IPUT array which defines a set of seven (7) Hollerith constants to be inserted in the converted number. The seven (7) Hollerith constants are:

IPUT (1) = '.', a decimal point
IPUT (2) = '-', a minus sign
IPUT (3) = '/', a slash
IPUT (4) = '%', a percent sign
IPUT (5) = ' ', a blank
IPUT (6) = '** TOO" } the number is
IPUT (7) = 'MUCH' } too big to convert

MOCTN: A variable used to index the position where the Hollerith constant is to be inserted in the output Hollerith string counting from right to left. In the case of a decimal number, the argument value passed to this variable will be N+1 where N is the decimal place of the number.

All parametric values have to be converted to integer numbers before they are passed to ITOH Subroutines for conversion into Hollerith code and insertion into ATLAS statement image because Hollerith code must be stored in integer array. Therefore, a parameter value of N decimal places has to be converted into an integer number by multiplying the parameter value by 10^N which will shift N decimal places to the left. This decimal to integer conversion is done in the IVT modules after all the parametric values are captured from the user. After the decimal to integer conversion, the integers are stored in CARY common array which will be passed to OTPT and ITOH for processing. Since the UNIVAC word size is 36 bits, the absolute value of an integer constant must be less than or equal to $2^{35} - 1 = 34,359,738,367$.

As an example to illustrate the process of capturing a parameter from the user in order to insert it into an ATLAS test program statement, assume a DC voltage of 12.5 volts is captured from the user in an IVT module. The following operations will be done to insert it into an ATLAS program statement in Hollerith code format.

(1) CARY (1) = 12.5 X 10 = 125
In the IVT module, convert 12.5 to an integer by multiplying by 10 and store it into CARY common array (assuming CARY (1)).

(2) Call ITOH (CARY (1), ICARD (6), 1, 1, 2)
In the OTPT module, call ITOH Subroutine to convert the 125 to its Hollerith Code equivalent with one decimal place and insert it into the ATLAS statement card image starting at card column 36. The conversion and insertion processes are achieved by ITOH Subroutine based on the information passed to it

by the arguments in the call statement where

CARY (1) = N: contains the integer 125 as the input number of ITOH Subroutine to be converted to its Hollerith code equivalent.

ICARD(6) = IOUT:

ITOH Subroutine will store the converted number (12.5) in Hollerith code in CARY (6) to be outputted back to the calling program. By storing the converted number in CARY (6) is effectively inserting its Hollerith code equivalent into the ATLAS statement card image starting at card column 36. ICARD is an array of 14 elements which holds the 80 columns card image. This is the entire ATLAS statement image. ICARD (6) will start at card column 36 since each UNIVAC word holds 6 characters.

1 = IDIM: I/O arrays, IN and IOUT, are dimensions to be 1 and go through the conversion loop once.

1 = KNST: Index of IPUT array IPUT (1) = '.' which is a decimal point to be inserted into the converted number.

2 = MOCTN: The position counting from right to left in the Hollerith code string for the decimal point to be inserted. In this case it will be the second place (125 $\rightarrow$ 12.5)

A flowchart of the ITOH Subroutine is shown in Figure 3.5.3-1 and a listing of the Subroutine with explanations is provided in AGEN II Software module listing.

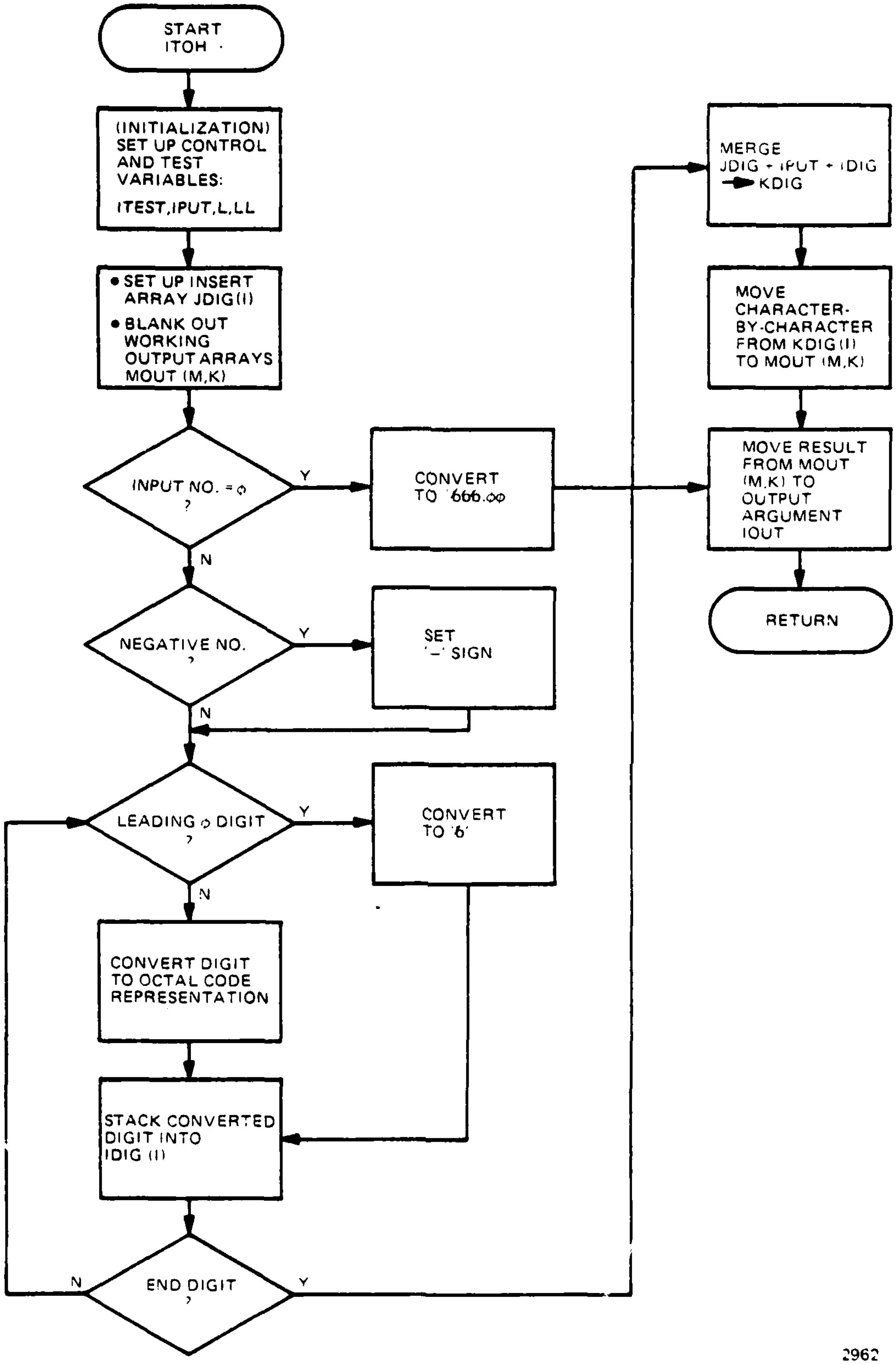


Figure 3.5.3-1. ITOH Subroutine Simplified Flowchart

3.5.4 MISR

The MISR is a multi-entry subroutine which organizes many basic I/O and file operations under one (1) module. There are four (4) entry points in the MISR routine and their functions are described below.

(1) OPEN

OPEN consists of a set of call subroutine statements to perform the files set-up functions as follows:

(a) Call the system routine NTRAN to rewind Drum 25 where a temporary scratch file is set up to store the ATLAS test program before editing for system routine NTRAN documentation (refer to FORTRAN V Library Manual Section 4.18).

(b) Call TNTO, a multi-entry UNIVAC Assembler routine, to open ATLAS Library file for reading.

(c) Call TNTO to set up temporary file TPF$ for storing the processed and edited ATLAS test program.

(d) Call TERR to print out the I/O error code if errors occured during the above I/O operations.

(2) READ

READ consists of a set of call subroutine statements to perform the READ functions as follows:

(a) Call TNTO to read statement by statement from the ATLAS Library File. The entire ATLAS statement is read into ICARD common array for further processing in the ITOH and OTPT routines.

(b) Call TERR to print out the I/O error code if errors occured during the read operation.

(3) WRIT

WRIT consists of a set of call subroutine statements to perform the WRITE function as follows:

(a) Call NTRAN to write the ATLAS statement image to the temporary file in Drum 25.

(b) Call TERR to print out the I/O error code if errors occured during the WRITE operations.

(c) If the argument WR=1 is passed to WRIT, the entire processed ATLAS statement will be displayed at the user terminal.

(4) WFLE

WFLE consists of a set of call subroutine statements to write and edit the processed ATLAS test program from the scratch file in Drum 25 to temporary TPF$ program file as follows:

(a) Call NTRAN to rewind the scratch file in Drum 25 for read.

(b) Call NTRAN to read the processed ATLAS statement from the scratch file into ICARD array.

(c) If the argument ED=1 is passed to WFLE, the ATLAS test program will be edited.

(d) Call TNTO to write the processed ATLAS statement from ICARD to temporary TPF$ program file.

(e) Call TNTO to close READ file.

(f) Call TNTO to close WRITE file.

(g) Call TERR to print out the I/O error code if errors occurred during the above file operations.

The functional difference between the scratch file in Drum 25 and temporary TPF$ program file is that each processed ATLAS test program is written sequentially in the TPF$ program file. while in Drum 25 the ATLAS programs are written by erasing the previous program in the same drum area,i.e., it is a scratch file. All the final ATLAS test programs are resident in the TPT$ program file.

3.5.5 OUTAPE

The OUTAPE is a FORTRAN subroutine used to control all the activities of converting the final ATLAS test programs (generated by AGEN II and residing in TPF$ program file) from Field Data to seven (7) bit ASCII and writing these converted ATLAS test programs onto a nine(9) track tape which is compatible to Data General's NOVA 3/Eclipse mini-computer. The resulting seven (7) bit ASCII tape is read into the Data General mini-computer by a program called MAGT which resides in the Data General mini-computer. However, the nine (9) track

seven (7) bit ASCII tape is needed only if the user runs AGEN II under the configuration in Figure 6-2 where no communication link exists between UNIVAC 1108 and Data General mini-computer and the resulting ATLAS test program tape must be transferred manually. Under the configuration in Figure 6-1 where communication link exists between the two (2) computers, the nine (9) track, seven (7) bit ASCII tape is not needed because the resulting ATLAS test programs are transferred to the Data General mini-computer by MODEM communication. Therefore, the execution of OUTAPE which generates the nine (9) track, seven (7) bit ASCII tape is made optional during AGEN II run. When the question of whether a test tape is needed comes up during AGEN II run, the user simply types 'N' (for No) at the console and the execution of OUTAPE will be skipped.

3.5.6 YNCHK

The YNCHK is a FORTRAN subroutine used to scan the 80 columns of the punched card for the first non-blank character and return it to the calling program for Yes or No check (Y or N check). The purpose of this subroutine is to provide the freedom for the user to type 'Y' or 'N' at any column within the 80 columns for "Yes" or "No" questions during AGEN II run.

3.5.7 ATOI

The ATOI is a FORTRAN subroutine used to convert the input data string from an "A" (alpha) format to an output in "I" (Integer) format to the calling program. The subroutine scans the 80 column card for the first non-blank "A" format character and converts all subsequent characters to integer until the first trailing blank.

The purpose of this subroutine is to provide the freedom for the user to input certain integer data start at any column within the 80 column card during the AGEN II run.

SECTION IV.

ATLAS TEST PROGRAM LIBRARY

The ATLAS Test Program Library contains the fully-structured, properly formatted and pre-tested ATLAS source test program skeletons for any combination of N and C codes for the five (5) analog networks currently implemented on AGEN II. The ATLAS Test Program Library is called "VITALLIBRARY#ANALOGNTWK$$." and it is resident in the UNIVAC 1108 at PRD. The ATLAS test program skeletons currently residing in the ATLAS Library is shown in Table 4-1. No numerical parametric (i.e., Input/Output) values are contained in these ATLAS test program skeletons. The parametric values are provided by the user or computed in their corresponding IVT modules. These parametric values are properly superimposed on their corresponding ATLAS test program skeleton by calling their corresponding OTPT sub-modules.

Each ATLAS test program skeleton is developed by a competent ATLAS language programmer based on the optimum test strategy and specifications supplied by a highly qualified test engineer. Each ATLAS test program skeleton is thoroughly debugged and tested for engineering, syntactic and semantic errors. Therefore, each ATLAS source program generated by AGEN II from the ATLAS Library is guaranteed to be of higher quality, greater uniformity and contain less errors than manually generated programs.

The ATLAS Test Program Library, VITALLIBRARY*ANALOGNTWK$$., may be accessed for changes or expansions. In the case of expanding AGEN II system by adding more characteristics into the existing networks or by adding additional networks with new characteristics, an ATLAS source program skeleton (thoroughly debugged and tested) corresponding to the new characteristics can be added to the ATLAS Test Program Library. The following card deck containing the required job control cards can be used to add a new ATLAS source program skeleton to the ATLAS Test Program Library under the UNIVAC 1108 EXEC 8 Operating System.

TO CHANGE OR ADD MODULES TO VITALLIBRARY

UNIVAC 1108 EXEC 8 COMMANDS	EXPLANATIONS
@RUN AGEN, PRD840, VITALLIBRARY	Run Card
@ASG, A VITALLIBRARY*ANALOGNTWK$$.	Assign the Library File
@COPY VITALLIBRARY*ANALOGNTWK$$.	Copy Library File to temporary file TPF$
@DELETE, SC .module	Delete the old module if existed
@ELT, ID .module	Read in the new module
(WORKING ATLAS SOURCE PROGRAM SKELETON CARD DECK)	
@END	End of Reading
@PACK	Pack the Library File
@ERS VITALLIBRARY*ANALOGNTWK$$.	Erase the old Library File
@COPY VITALLIBRARY*ANALOGNTWK$$.	Copy the updated Library File from TPF$ back to the Library File
@LIST, @. ANALOGNTWK$$. module	Get a source listing of the new module
@FIN	
##	Finish

NOTE: Module is the name of the ATLAS source program skeleton

A sample of an ATLAS test program skeleton is shown in Figure 4-1.

```
C       OSCILLATOR FREQUENCY STABILITY TEST.FILE NAME 'OSCFS' $
        DECLARE DECIMAL, 'MTIME', 'FO','Y','FMAX' $
C           CCCCC1                                                      $
        DEFINE 'VDC',                     12.0                          $
        DEFINE 'TIME',                    12.0                          $
        DEFINE 'FUL',                   1120.0                          $
        DEFINE 'FLL',                    880.0                          $

        DEFINE 'VMAX',                    12.0                          $
        DEFINE 'VTHRSH',                   1.2                          $
        DEFINE 'FCHOICE',                    1                          $
C       PROCEDURAL SECTION $
E  2000 DISPLAY,"PERFORM HOOKUP AS FOLLOWS.PRESS PROCEED WHEN COMPLETE.
                 ID-BOX            TABLE-TOP
                 UUT PWR           J29
                 UUT O/P           J37
                 INSERT UUT INTO THE UUT CONNECTOR ON THE ID-BOX." $
        WAIT-FOR MANUAL-INTERVENTION $
        RECORD,"          "

        RECORD'FUL',"UPPER LIMIT= #########.#HZ" $
        RECORD'FLL',"LOWER LIMIT= #########.#HZ" $
        RECORD,"          " $
        'Y'= 0  $
        'MTIME' = 'TIME' $
        APPLY DC-SIGNAL DC2A, VOLTAGE 'VDC'V $
        DELAY 100 MSEC $
        'FMAX'= 'FUL'* 1.5 $
   2010 COMPARE 'FCHOICE', GT 1.5 $
        GOTO STEP 2015 IF GO $
C       MEASUREMENT IF UUT FREQ RANGE 2HZ-1MHZ. $
        MEASURE(FREQ 'FO'HZ), AC-SIGNAL, FREQ MAX 'FMAX'HZ,
        VOLTAGE MAX 'VMAX'V, AC-COUPLE, THRESHOLD 'VTHRSH'V, CNX BNC 1 $

        GOTO STEP 2017 $
C       MEASUREMENT IF UUT FRE RANGE 60KHZ-100 MHZ $
   2015 MEASURE(FREQ'FO'HZ),AC-SIGNAL,FREQ MAX'FMAX'HZ,VOLTAGE MAX'VMAX'V,
        AC-COUPLE, THRESHOLD 'VTHRSH'V, TEST-EQUIP-IMP 50OHM, CNX BNC 1 $
   2017 RECORD'FO',"MEASURED VALUE = #########.# HZ" $
        'Y'='Y'+1 $
        COMPARE 'FO', UL 'FUL' LL 'FLL' $
        GOTO STEP 2020 IF NOGO $
        DELAY 1.0 SEC $
        'MTIME' = 'MTIME' - 5.0 $
        COMPARE 'MTIME', LE 0 $
        GOTO STEP 2010 IF NOGO $
        RECORD,"UUT PASSED FREQUENCY STABILITY TEST" $
        GOTO STEP 2030 $
   2020 RECORD,"UUT FAILED FREQUENCY STABILITY TEST" $
   2030 REMOVE ALL $
        FINISH $
        TERMINATE $
C EOT     $
```

FIGURE 4-1   ATLAS TEST PROGRAM SKELETON SAMPLE

NETWORK	CHARACTERISTIC	ATLAS TEST MODULE NAME
AMPLIFIER	AMPLITUDE RESPONSE	AMPAR
	LINEARITY	AMPLN
OSCILLATOR	FREQUENCY STABILITY	OSCFS
	AMPLITUDE STABILITY	OSCAS
POWER SUPPLY	REGULATION	PWSREG
	RIPPLE	PWSRPL
	OVERCURRENT	PWSOVC
MIXER	ISOLATION	MIXISO
	FREQUENCY RESPONSE	MIXFRS
	CONVERSION LOSS	MIXCON
FILTER	INSERTION LOSS	FILILO
	PHASE RESPONSE	FILPHR
	INPUT IMPEDANCE	FILINZ
	OUTPUT IMPEDANCE	FILOUZ
	AMPLITUDE RESPONSE	FILAMR

TABLE 4-1 ATLAS Test Module Residing in ATLAS Test Program Library

SECTION V

EXPANDING AGEN II SOFTWARE SYSTEM

One salient feature of the design of AGEN II Software System is the provision of easy expansion in the future with minimum impact on the current system. AGEN II is comprised of two (2) types of software, the FORTRAN executive software and the ATLAS Test Programs residing in the ATLAS Library. Making expansions in either area is relatively simple. Expanding the ATLAS Test Program Library is described in Section IV, and AGEN II Software Modules Listing Manual contains the FORTRAN source code listings to enable programmer to do the expansions and modifications. The methods and details of adding another network to AGEN II System are best illustrated by an example. Assume that another network with three (3) characteristics is going to be added to the existing AGEN II System. This network could be represented by N=6 (the 6th network in AGEN II), C=1 for its first characteristic, C=2 for its second characteristic and so on and the expanded AGEN II System will be created as shown in Figure 5-1 with the existing five (5) networks plus the addition of the 6th network with three (3) characteristics and the three (3) corresponding ATLAS test programs added into the ATLAS Test Program Library. This hypothetical added network (assume to be attenuator) is already integrated into AGEN II System except with its corresponding IVT 6 and OTPT 6 implemented in program stubs (dummy programs to be substituted by actual programs) and without its corresponding ATLAS test programs in the ATLAS Test Program Library. In fact, AGEN II System is already implemented for N=6 to 8 with their IVT's and OTPT's in the form of program stubs. After the program stubs are integrated, the real tasks are to create the corresponding actual programs, namely IVT's and OTPT's, for their substitutions and to create the corresponding ATLAS test programs in the ATLAS Test Program Library. Therefore, there are no needs to modify AGEN, DNSC, and BBAS module to accommodate these networks. For the sake of completeness, the expansion of AGEN II System to include the 6th network is described in details in the following sections.

5.1 AGEN EXPANSION

Minor modifications can be made in the AGEN main control program for the addition of a 6th network in to the existing AGEN II System. The modifications are to expand two(2) computed GO TO statements and to add two(2) subroutine call statements and two (2) unconditional GO TO statements. One subroutine call statement is to call IVT 6 for parametric data entry and the other is to call OTPT 6 for writing the user requested ATLAS test programs.

The FORTRAN codes related to calling IVT6 for parametric data entry are expanded as follows:

```
      c ------------------------------------
      c ------GO to individual IVT routine depending on the N code
      c ------------------------------------
              GO to (281, 282, 283, 284, 285, XXX), NCD
        281   Call IVT1                @ for N code = 1
              GO to 290
         :      :                           :
        285   Call IVT5                @ for N code = 5
              GO to 290
        XXX   Call IVT6                Inserted statements for
              GO to 290                Calling IVT6 and exit
```

where NCD (obtained from DNSC) contains the N code for the computer GO TO statement and XXX is the expanded statement of entry if NCD contains 6. If systematic statement numbering is used, XXX should be numbered 286 because this group of call statements has the general numbering format of 28N where N is the N code.

The FORTRAN codes related to calling OTPT6 at its multi-entry points for writing ATLAS test programs of user requested characteristics are expanded as follows:

```
      c ------------------------------------
      c ------GO to individual OTPT routine depending on the N and C code
      c ------------------------------------
              GO to (410, 420, 430, 440, 450, YYY), NCD
        410   Continue
              WRITE (6, 71)            (Printout network   being processed)
              GO TO (411, 412), CCD    @ for N code = 1
        411   Call OTPT11
              GO TO 550
        412   Call OTPT12
              GO TO 550
                :
        YYY   Continue
              WRITE (6, ZZ)            Inserted statements for calling
              GO TO (iii,jjj,kkk),CCD  OTPT6 at its three entry points
```

```
iii    Call OTPT61        and exit for its three
       GO TO 550          characteristics
jjj    Call OTPT62
       GO TO 550
kkk    Call OTPT63
       GO TO 550
```

where NCD contains the N code for the first computed GO TO statement and YYY is the expanded statement of entry if NCD contains 6. CCD (obtained from DNSC) contains the C code for the second computed GO TO statement and iii, jjj, kkk are the statement of entry points for the three characteristics depending on the value of CCD. If systematic statement numbering is used, the statements should be numbered as follows:

STATEMENT	NUMBER ASSUMED	GENERAL FORMAT
iii	461	4NC
jjj	462	4NC
kkk	463	4NC
YYY	460	4NO
ZZ	76	7N

where N is the N code of the network and C is the C code of the characteristic.

5.2 DNSC Expansion

The only modification needed in the DNSC module to accommodate an added network is the N (Network) Code Table which defines all the networks in the AGEN II System. The modification involves the addition of one (1) line per added network and the expansion of the size of the table.

The expanded N Code Table is shown below.

```
      DATA ((NCTB(I,J),J=1,5),I=1,9)    @ N CODES
     * /8,0,0,0,0
     * ,'    1        AMPLIFIERS             '
     * ,'    2        OSCILLATORS            '
     * ,'    3        POWERS SUPPLIES        '
     * ,'    4        MIXERS                 '
     * ,'    5        FILTERS                '
     * ,'    6        ATTENUATORS            '
     * ,'    7        DEMODULATORS           '
     * ,'    8        PULSE AMPLIFIERS       '/
```

The N Code Table is dimensioned I by J (6 characters) where J (1 to 5) is the number of UNIVAC words per line which will store a maximum of 30 characters and I (1 to 9) is the number of networks (8) defined in the table plus one (1) line (line 18 which specifies 8 networks in the table). For the hypothetical added network (attenuators), N=6, line 24 must be added. If the 9th network is going to be added in the AGEN II System, the size of the N Code Table must be expanded by changing I=1, 10 on line 17, * /9, 0, 0, 0, 0 on line 18 and add line 27 at the end of the table to define the 9th network (N=9). Note that N=6 to 8 are the hypothetical networks already integrated into the AGEN II System with their IVT's and OTPT's in the form of program stubs as described in the beginning of the section.

5.3 BBAS Expansion

The only modification needed in the BBAS FORTRAN procedure to accommodate an added network is the C (characteristics) Code Table which defines the characteristics of all the networks in the AGEN II System. The modification involves the addition of the C Code Table of the added network and the expansion of the size of the table. The C Code Table of the hypothetically-added network (Attenuators, N=6) with three (3) characteristics must be added in BAAS procedure as shown below.

```
60:        DATA ((CCTB(I,J,6),J=1,8),I=1,9)       @ C CODE FOR N = 6
61:      *  /3,0,0,0,0,0,0,0
62:      *  , 1, 'ATTAMR       ', 'AMPLITUDE RESPONSE                '
63:      *  , 2, 'ATTINZ       ', 'INPUT IMPEDANCE                   '
64:      *  , 3, 'ATTOUZ       ', 'OUTPUT IMPEDANCE                  '
65:      *  , 0, 7*'       '
66:      *  , 0, 7*'       '
67:      *  , 0, 7*'       '
68:      *  , 0, 7*'       '
69:      *  , 0, 7*'       '/
```

The C Code Table is dimensioned I x J x N where J (1 to 8) is the number of UNIVAC words per line which will store a maximum of 48 characters, I (1 to 9) is the maximum number of characteristics (8) that the current table will be able to define plus one (1) line (line 61 which specifies 3 characteristics currently defined in the table), and N=6 is the N code of the added network. If there are 9 characteristics in this network, the size of this C Code Table must be expanded by changing I=1, 10 on line 60, * /9, 0, 0, 0, 0, 0, 0, 0 on line 61 and add line 70 at the end of the table to define the 9th characteristic (C=9)

5.4 IVT Expansion

For each network added to the AGEN II System, an IVT module must be created to capture the parametric values from the user and to calculate the expected outputs for the user requested characteristics. The IVT module must cover all specified characteristics of the network. The top-down implementation method is therefore suggested. This method starts at the top by designing the IVT modules in modular program stubs. Then, while the engineering analysis and specification of the additional network are still under study, the program changes in AGEN, DNSC, BBAS as described previously may proceed. When the engineering specifications are available, the actual coding of the IVT modules becomes a substitution of the program stubs. The systematic nomenclature of the IVT modules (as discussed in Section 3.3) in the form of IVTN, where N is the N code for the added network, is also suggested. As for the 6th hypothetically-added network, IVT will be created to capture the parametric values and to calculate the expected outputs for its three (3) specified characteristics.

5.5 OTPT

For additional networks, an OTPT module must be created to write the ATLAS source test programs for the user-requested characteristics. The OTPT module must cover all specified characteristics of the network. Again the top-down implementation method is suggested. The multi-entry structure (one entry point per characteristic) and the systematic nomenclature of the OTPT modules (as discussed in Section 3.4) in the form of OTPTN and OTPTNC for the entry points are also suggested. As for the 6th hypothetically-added network, OTPT6 with three (3) entry points for its three (3) characteristics, OTPT61, OTPT62, OTPT63, will be created to write their corresponding ATLAS source test programs.

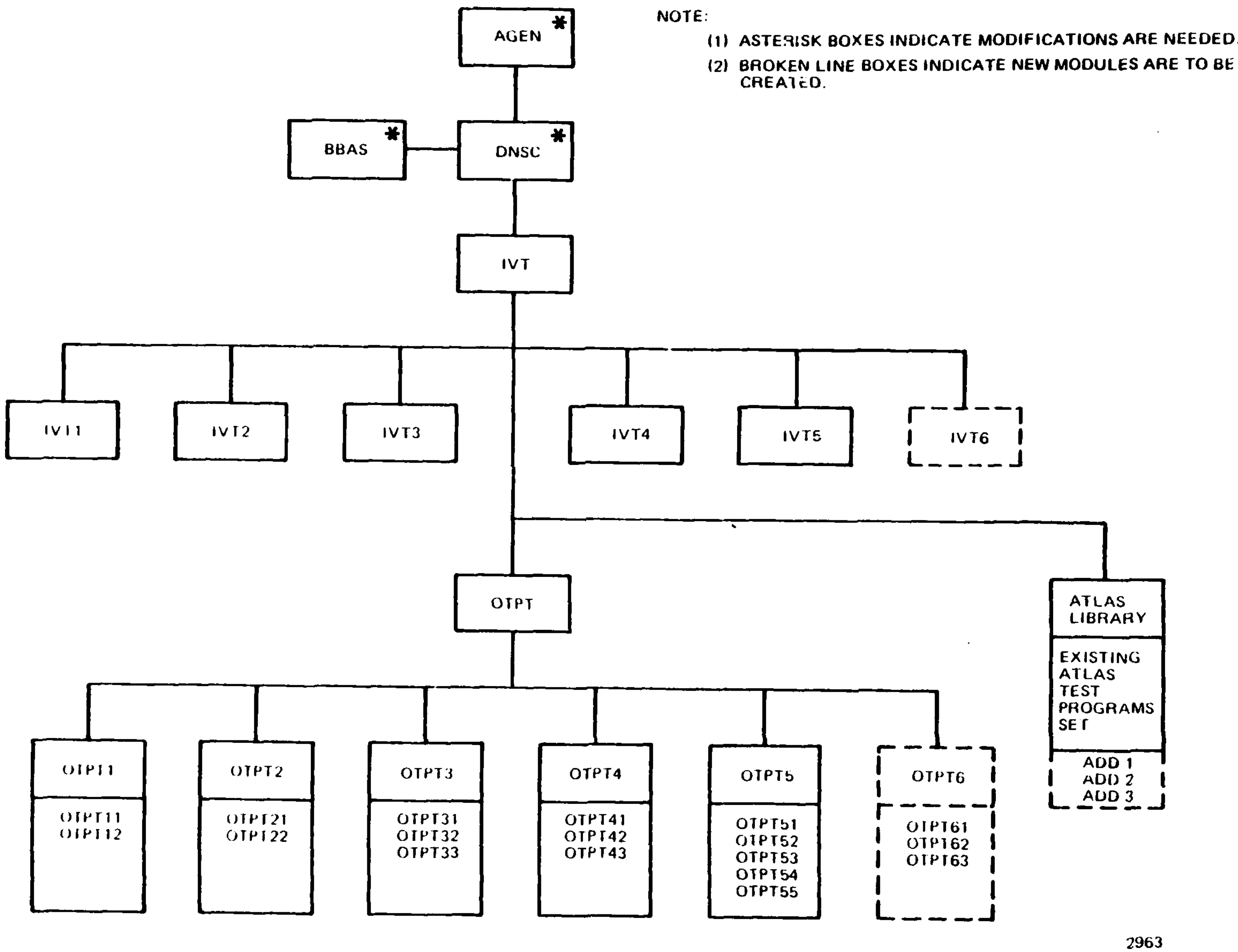


Figure 5-1. AGFN II Expanded Software Family Tree

SECTION VI

6.0 THE INTEGRATED AGEN II TEST SYSTEM - ATLAS LIBRARY, AGEN II, EQUATE

The ATLAS Source Program Library, the AGEN II Executive Software System, and the EQUATE Test Station are the three (3) major elements comprising the integrated AGEN II test system. Each major element performs a distinctive function. The ATLAS Source Program Library stores all the ATLAS test program skeletons in source form and supplies to AGEN II System for program generation. The AGEN II Executive Software System generates ATLAS source test programs free of syntactic, semantic, and engineering errors for functional testing of analog circuits on the EQUATE Test Station. The EQUATE Test Station compiles the ATLAS Source Test Programs generated by AGEN II and tests the UUT.

The integrated AGEN II test system may function under two (2) configurations, with or without communication link as shown in Figure 6-1 and Figure 6-2 respectively. With the communication link (Figure 6-1), the Data General NOVA 3/ECLIPSE mini-computer which controls the EQUATE Test Station is linked to UNIVAC 1108 installation at PRD Electronics where AGEN II software system resides. Under this configuration, the user may run AGEN II from a remote site by inputting parametric values through the terminal of a Data General NOVA 3/ECLIPSE mini-computer. The ATLAS source test programs generated by AGEN II are sent directly to the disk of the Data General mini-computer by communication lines. After the execution of AGEN II, the ATLAS source test programs for the user requested characteristics are present in the Data General mini-computer disk file. The user may compile the ATLAS source test programs by calling the ATLAS compiler in EQUATE and then proceed to test the UUT's.

Under the configuration without communication link between UNIVAC 1108 and Data General NOVA 3/ECLIPSE mini-computer (Figure 6-2), a tape containing the ATLAS source test programs must be generated. The ATLAS test programs source tape must be manually transferred and loaded into the Data General NOVA 3/ECLIPSE mini-computer. Then the compilation of the ATLAS Source Program and testing of the UUT's may proceed in the EQUATE station. Operation under this configuration is possible only if both the UNIVAC 1108 and EQUATE Test Station are resident in the same site. This is, of course, due to the restriction of the movability of the ATLAS source tape.

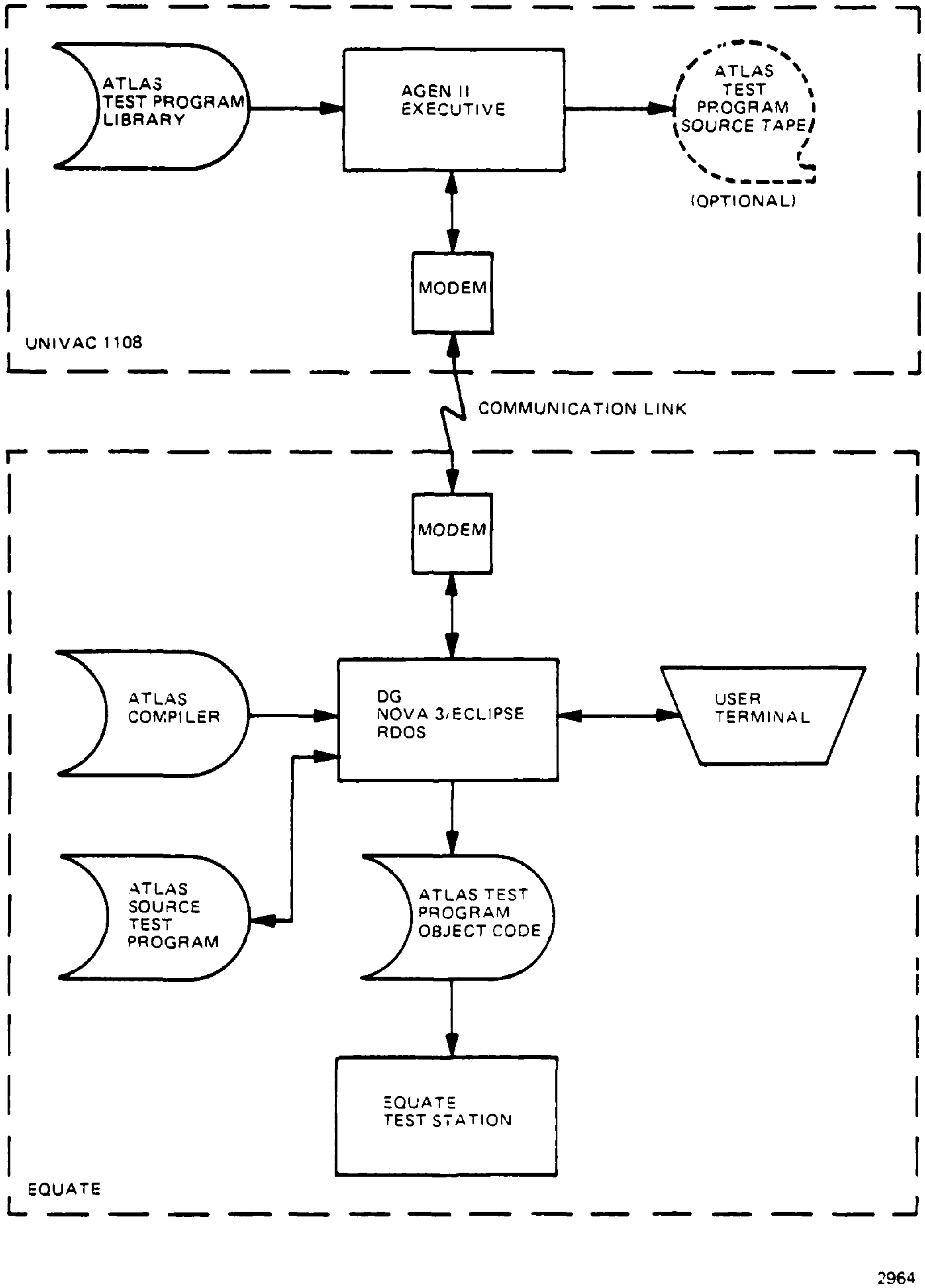


Figure 6-1. The Integrated Test System - With Communication Link

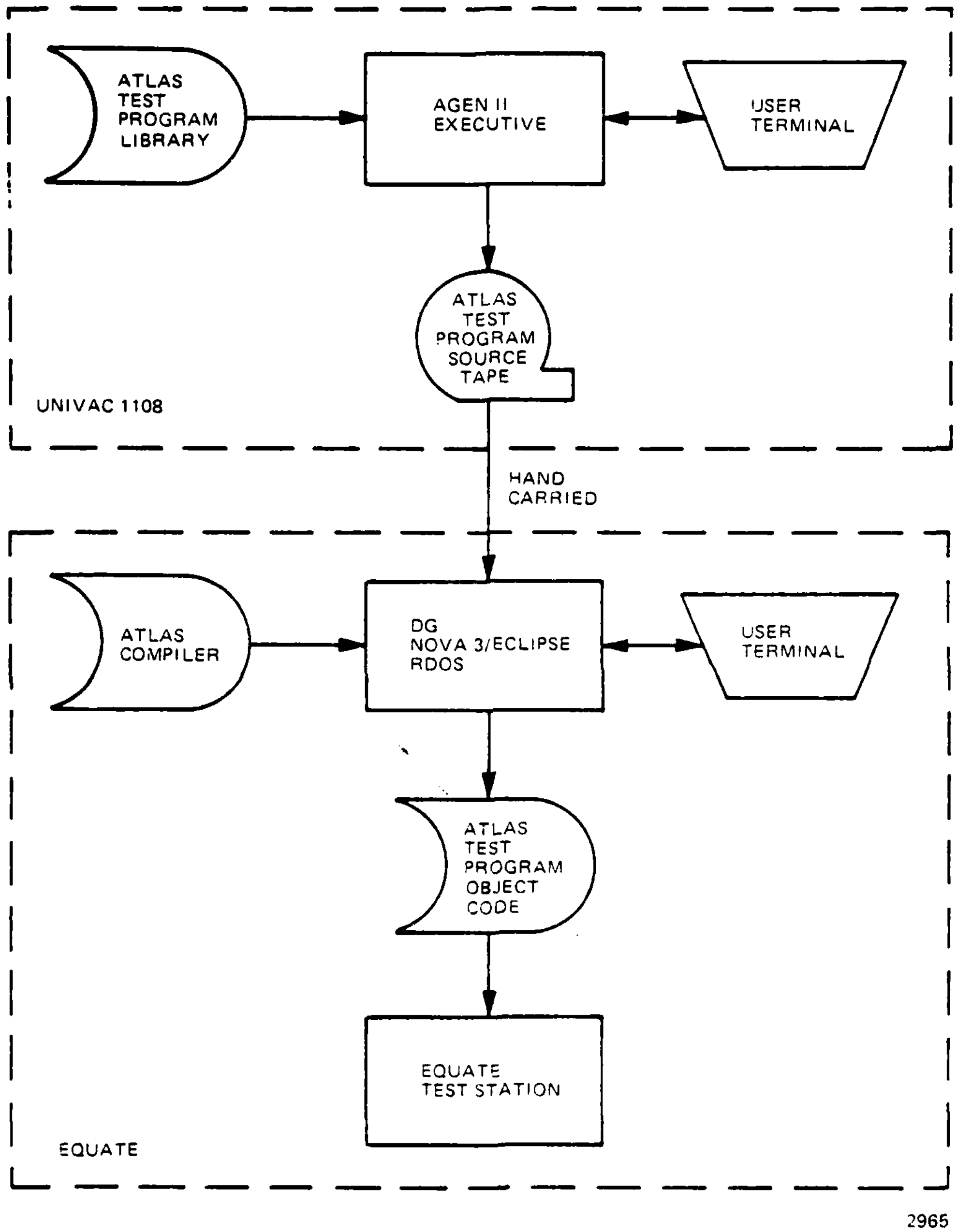
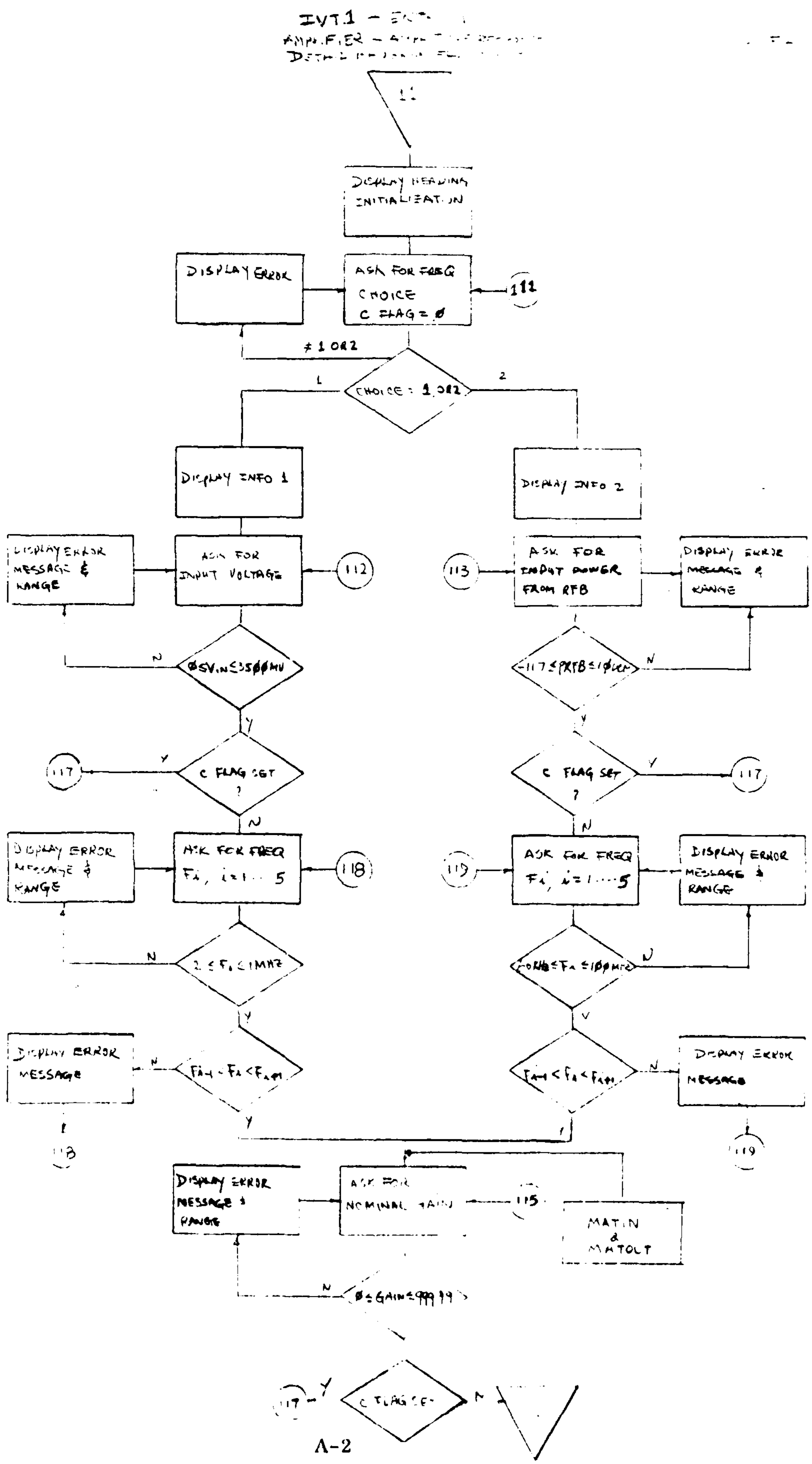
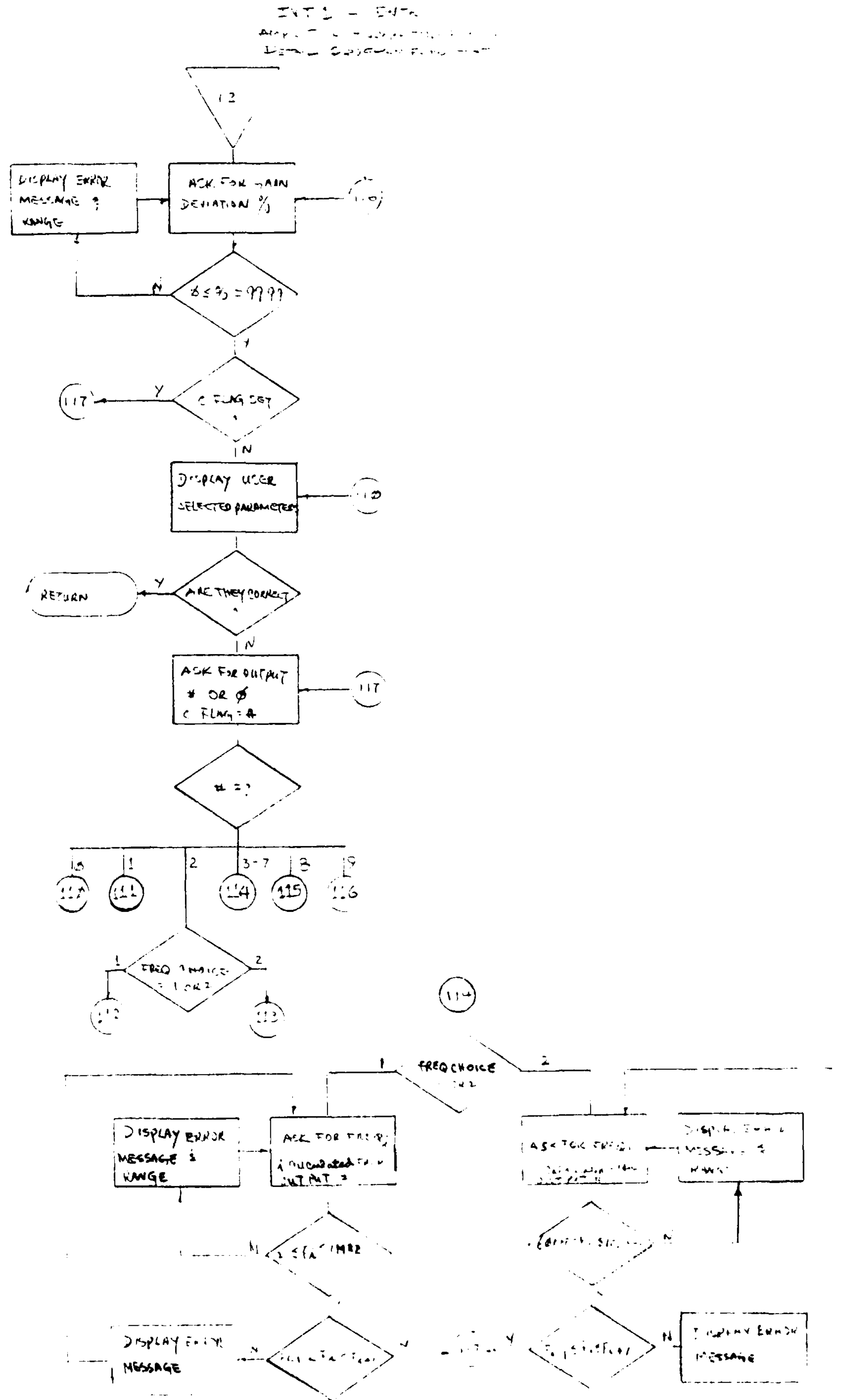


Figure 6-2. The Integrated Test System - Without Communication Link

APPENDIX A

This Appendix contains all the flowcharts of the IVT Modules for the five (5) analog networks implemented in AGEN II. Each characteristic has a flowchart showing the process of capturing the parameters and tolerances and their validation. After each flowchart, there is an Array Structure Table showing where the parametric values are being stored. These parametric values are stored in the CARY Common Array in the integer form and are passed to their corresponding OTPT modules to be superimposed on the ATLAS test program skeletons.


IVT1
DISPLAY HEADING INITIALIZATION
DISPLAY ERROR
ASK FOR FREQ CHOICE C FLAG = Ø
111
≠ 1 OR 2
CHOICE = 1, OR 2
DISPLAY INFO 1
DISPLAY INFO 2
DISPLAY ERROR MESSAGE & RANGE
ASK FOR INPUT VOLTAGE
112
113
ASK FOR INPUT POWER FROM RFB
C FLAG SET ?
117
DISPLAY ERROR MESSAGE & RANGE
ASK FOR FREQ
118
119
DISPLAY ERROR MESSAGE
ASK FOR NOMINAL GAIN
115
MATIN & MATOUT
C FLAG SET



12
DISPLAY ERROR MESSAGE & RANGE
ASK FOR GAIN DEVIATION %
N
Y
C FLAG SET
DISPLAY USER SELECTED PARAMETERS
RETURN
ARE THEY CORRECT
ASK FOR OUTPUT # OR Ø
C FLAG = Ø
117
1
2
3-7
8
9
111
114
115
116
FREQ CHOICE
114
DISPLAY ERROR MESSAGE & RANGE
ASK FOR FREQ
DISPLAY ERROR MESSAGE
DISPLAY ERROR MESSAGE

ARRAY STRUCTURE

NETWORK: AMPLIFIER (N = 1)

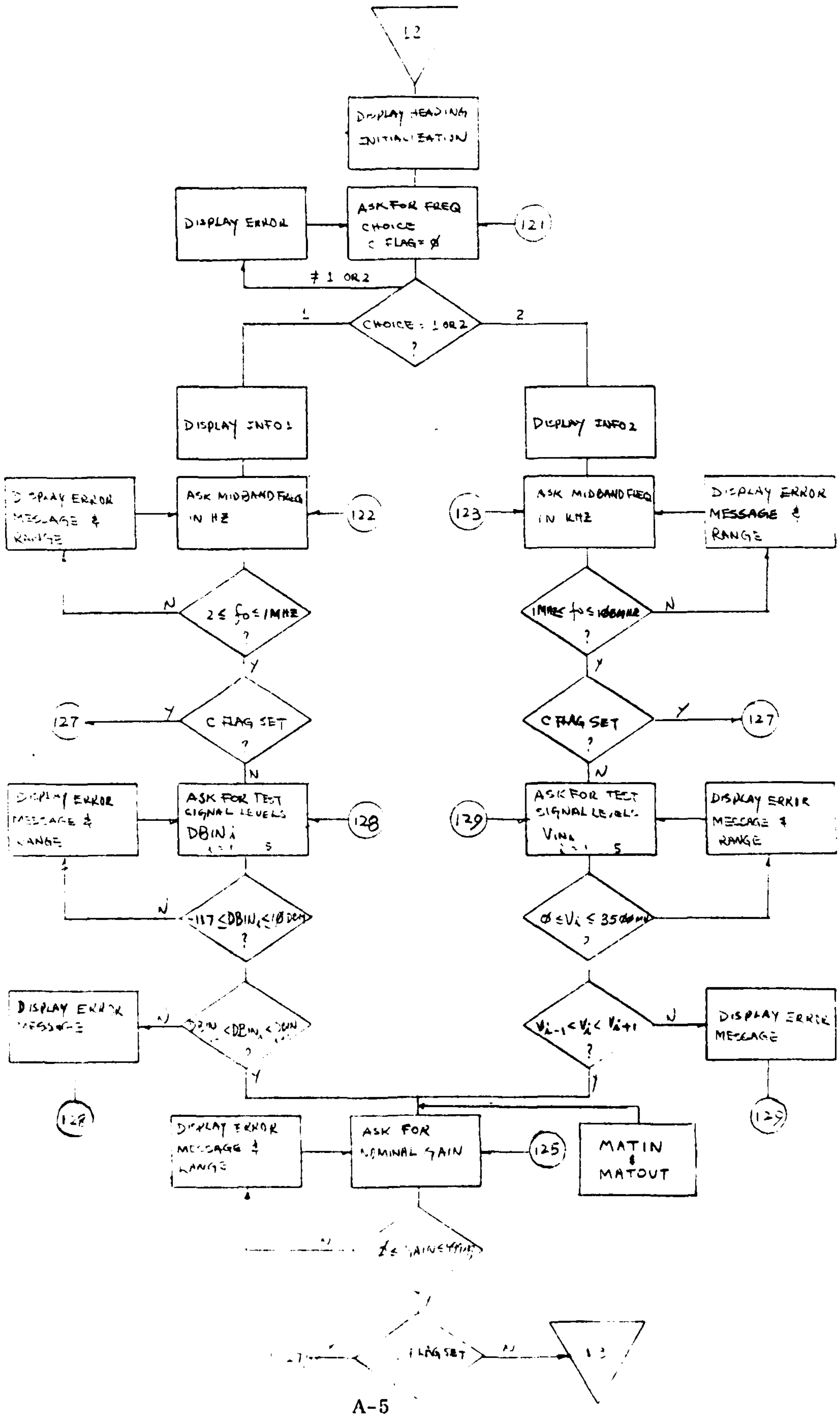
CHARACTERISTIC: AMPLITUDE RESPONSE (C = 1)

ARRAY NAME: CARY

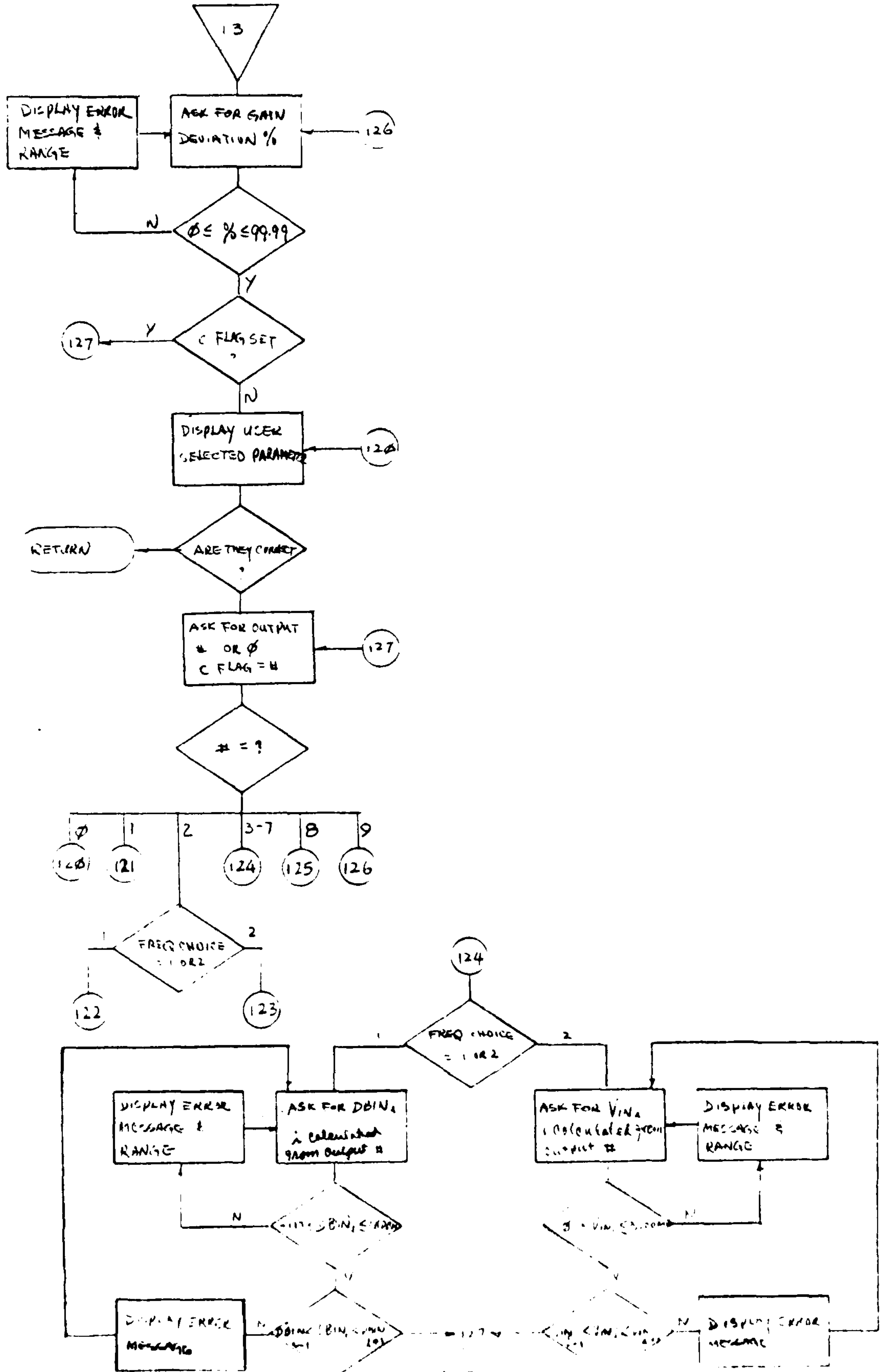
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT11 MODULE.

CARY(N)	VARIABLE	EXPLANATION
1 – 5	FREQ(5)	FREQUENCIES.
6	FC	FREQUENCY CHOICE.
7	INVOLT	INPUT VOLTAGE FROM GENERATOR.
8	IPREB	INPUT POWER FROM RFB.
9	NGDP	NOMINAL GAIN DEV. %
10	NOMGN	NOMINAL GAIN
11	—	NOMINAL GAIN UPPER LIMIT
12	—	NOMINAL GAIN LOWER LIMIT.
13	VDC	VOLTAGE FOR UUT
14	MATIN	INSERTION LOSS INPUT MATCHING NETWORK DB
15	MATOUT	INSERTION LOSS OUTPUT MATCHING NETWORK DB

DETAIL PROGRAM FLOWCHART



DETAIL PROGRAM FLOWCHART



AGEN
ARRAY STRUCTURE

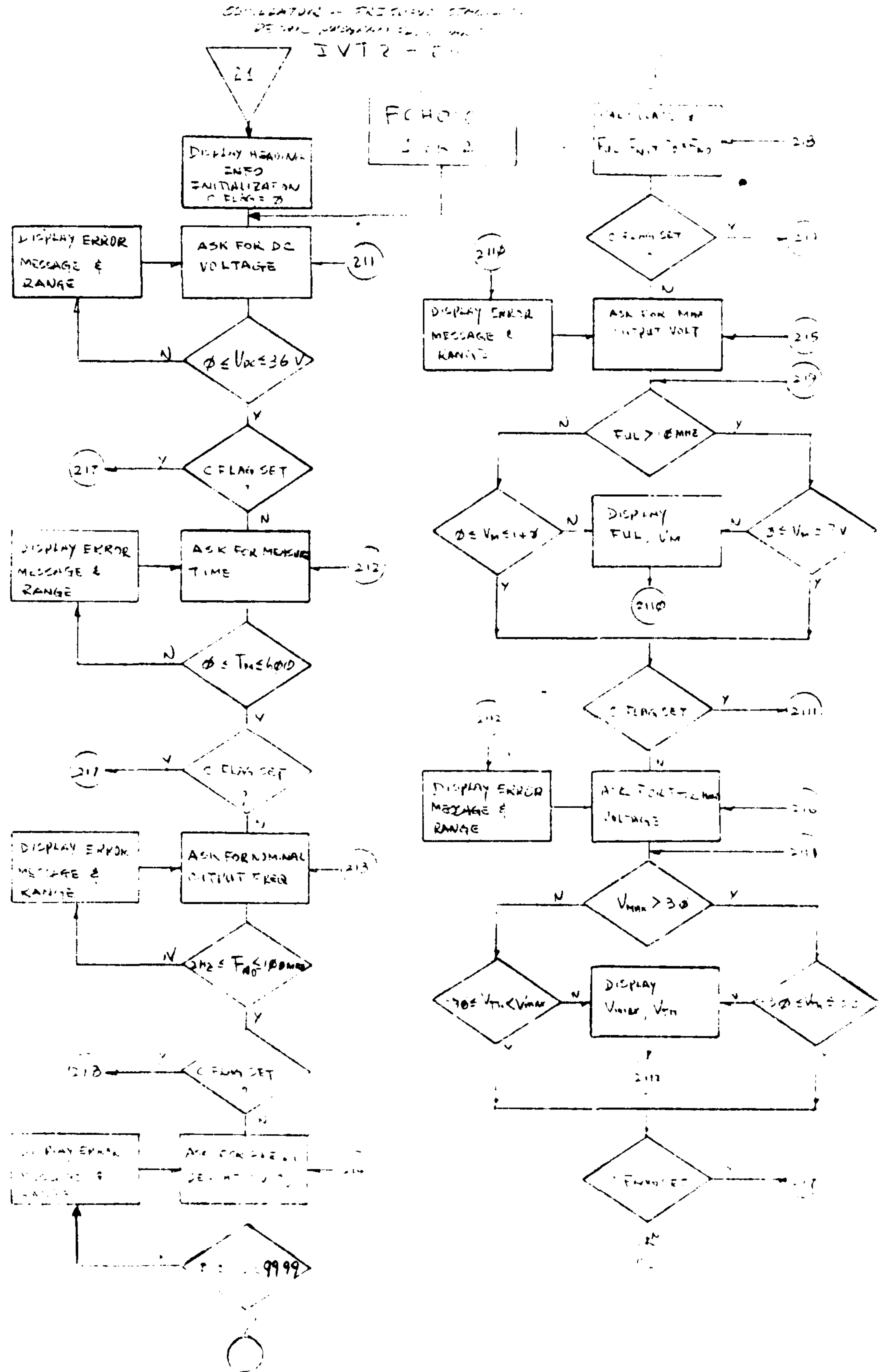
NETWORK: AMPLIFIER (N=1)

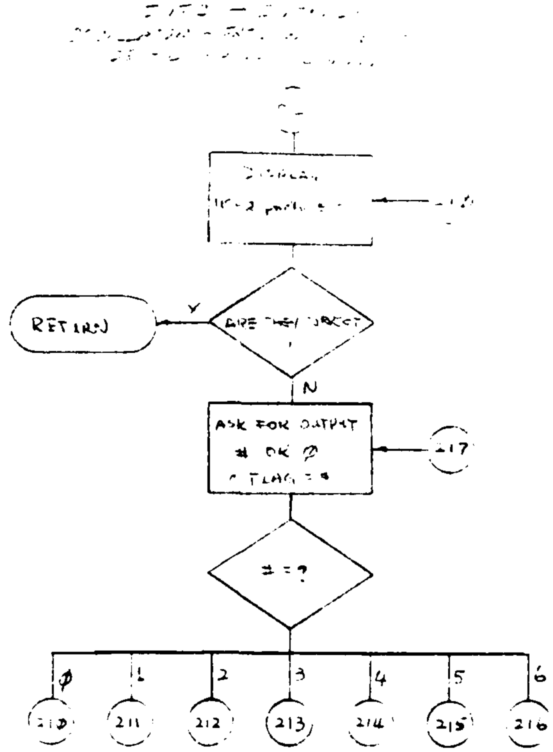
CHARACTERISTIC: LINEARITY (C=2)

ARRAY NAME: CARY

PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT12 MODULE

CARY(N)	VARIABLE	EXPLANATION
1	FC	FREQUENCY RANGE CHOICE
2	MIDFRQ	MIDBAND FREQUENCY.
3-7	DBIN(5)	5 TEST SIGNAL LEVELS FOR FC = 1.
8-12	VIN(5)	5 TEST SIGNAL LEVELS FOR FC=2.
13	NOMGN	NOMINAL GAIN AT MIDBAND FREQ.
14	NGDP	NOMINAL GAIN DEVIATION %.
15	—	NOMINAL GAIN UPPER LIMIT.
16	—	NOMINAL GAIN LOWER LIMIT.
17	VDC	DC VOLTAGE TO UUT
18	MATIN	insertion Loss Input matching Network DB
19	MATOUT	insertion Loss output matching Network DB

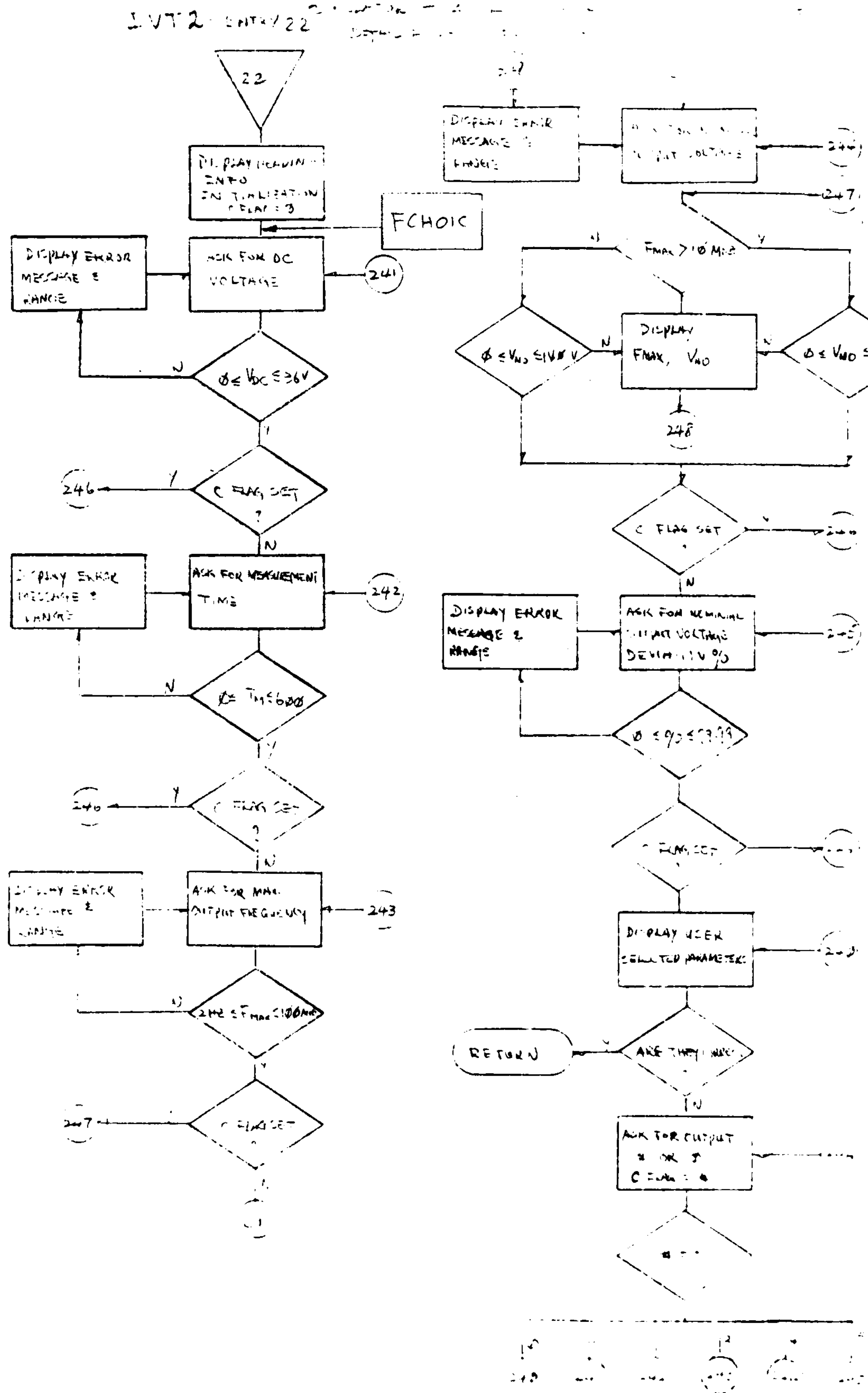

IVT2
21
DISPLAY HEADING INFO INITIALIZATION C FLAG = Ø
ASK FOR DC VOLTAGE
DISPLAY ERROR MESSAGE & RANGE
211
C FLAG SET
ASK FOR MEASURE TIME
212
ASK FOR NOMINAL OUTPUT FREQ
213
218
9999
DISPLAY FUL, VM
DISPLAY ERROR MESSAGE & RANGE
ASK FOR MAX OUTPUT VOLT
215
DISPLAY VMAX, VM
Y
N


RETURN
ARE THEY CORRECT
Y
N
ASK FOR OUTPUT # OR Ø
217
= ?
Ø
1
2
3
4
5
6
210
211
212
213
214
215
216

AGEN
ARRAY STRUCTURE

NETWORK: OSCILLATOR (N=2)
CHARACTERISTIC: FREQUENCY STABILITY (C=1)
ARRAY NAME: CARY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT21 MODULE.

(N)	VARIABLE	EXPLANATION
1	VDC	DC VOLTAGE
2	MT	MEAS. TIME FOR FREQ. STABILITY TEST.
3	NOF	NOMINAL OUTPUT FREQ.
4	FDP	FREQ. DEVIATION %
5	VM	MAX OUTPUT VOLTAGE
6	VTH	INPUT THRESHOLD VOLTAGE.
7	FUL	UPPER LIMIT FOR FREQUENCY.
8	—	LOWER LIMIT FOR FREQUENCY
9	FCHOIC	Frequency choice 1 or 2 1) 2 HZ - 1 MHZ 2) 60 KHZ - 100 MHZ

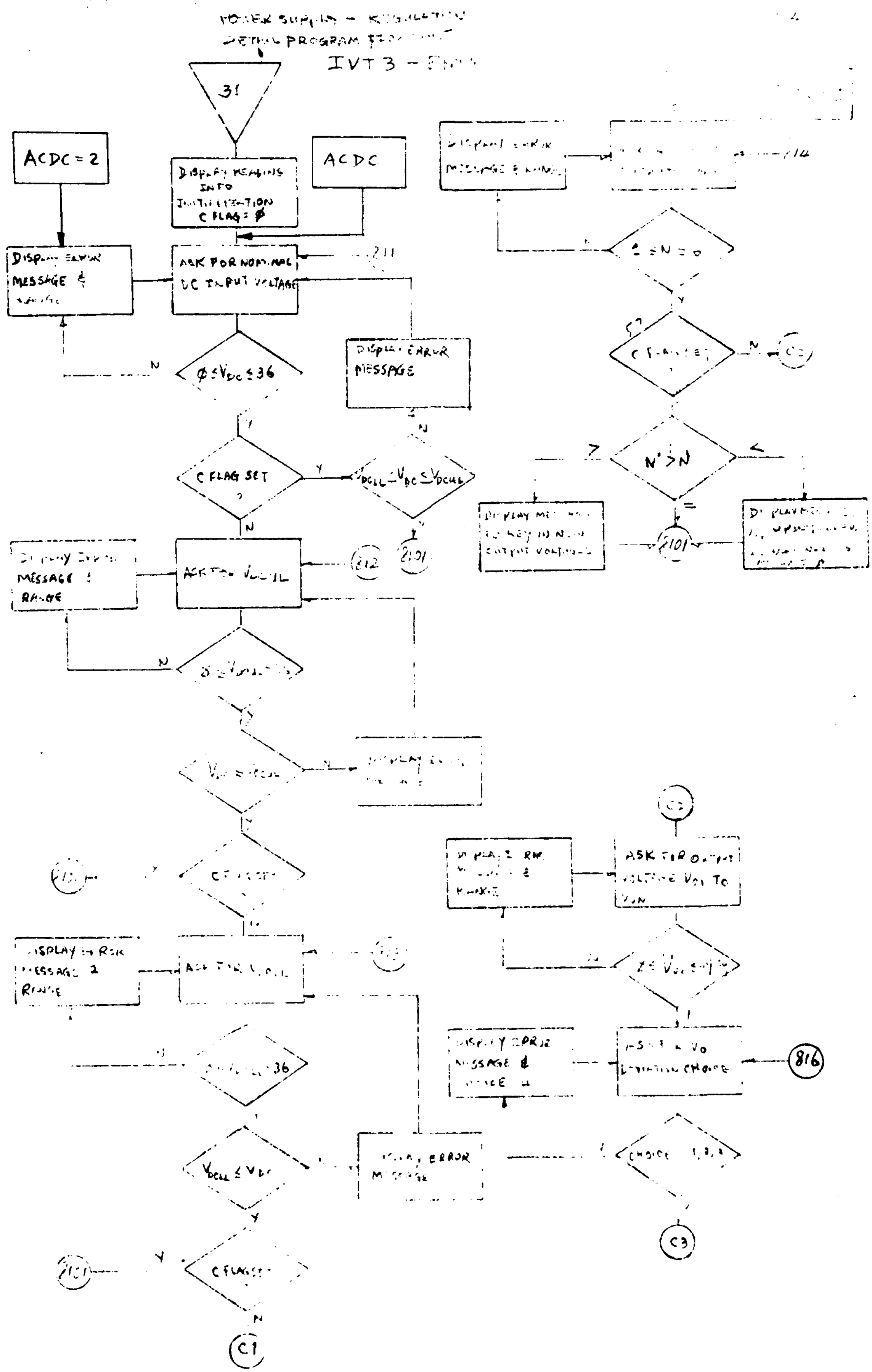


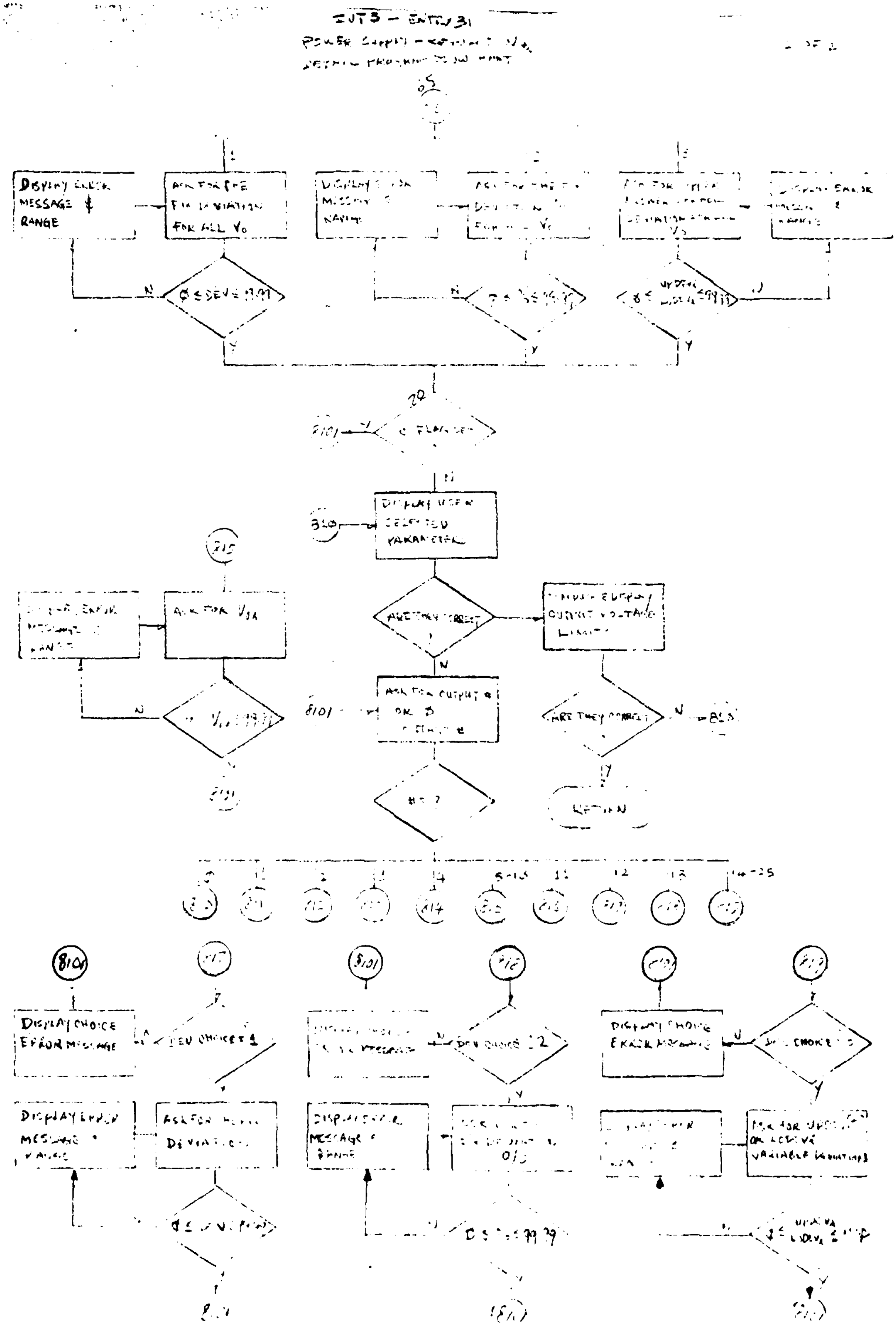
22
FCHOIC
ASK FOR DC VOLTAGE
241
DISPLAY ERROR MESSAGE & RANGE
C FLAG SET ?
246
ASK FOR MEASUREMENT TIME
242
DISPLAY ERROR MESSAGE & RANGE
243
DISPLAY FMAX, VNO
248
C FLAG SET
DISPLAY ERROR MESSAGE & RANGE
DISPLAY USER SELECTED PARAMETERS
RETURN
ARE THEY OK?
ASK FOR OUTPUT

ARRAY STRUCTURE

NETWORK: OSCILLAT?R (N=2)
CHARACTERISTIC: AMPLITUDE STABILITY (C=2)
ARRAY NAME: CAKY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT22

(N)	VARIABLE	EXPLANATION
1	DCVOLT	DC VOLTAGE.
2	TMEAS	MEASUREMENT TIME FOR AMPLITUDE STABILITY TEST.
3	MAXOF	MAX OUTPUT FREQUENCY.
4	NOMOV	NOMINAL OUTPUT VOLTAGE.
5	NOVDP	NOMINAL OUTPUT VOLTAGE DEV. %
6	—	NOMINAL OUTPUT VOLTAGE UPPER LIMIT
7	—	NOMINAL OUTPUT VOLTAGE LOWER LIMIT.
8	FCHOIC	Frequency CHoice 1 or 2 1) 2 HZ - 1 MHZ 2) 60 KHZ - 100 MHZ


POWER SUPPLY - REGULATION
DETAIL PROGRAM
IVT3
31
ACDC=2
ACDC
DISPLAY HEADINGS INFO INITIALIZATION C FLAG = Ø
DISPLAY ERROR MESSAGE & RANGE
ASK FOR NOMINAL DC INPUT VOLTAGE
211
Ø ≤ V_DC ≤ 36
DISPLAY ERROR MESSAGE
C FLAG SET ?
ASK FOR V_DCUL
812
2101
C FLAG SET
ASK FOR OUTPUT VOLTAGE V_O1 TO V_ON
816
C1
C3
N'>N
DISPLAY ERROR MESSAGE & RANGE
ERROR MESSAGE


IJTS - ENTRY 31
DISPLAY ERROR MESSAGE & RANGE
ASK FOR THE FIX DEVIATION FOR ALL Vo
DISPLAY USER SELECTED PARAMETER
ARE THEY CORRECT
ASK FOR OUTPUT #
ARE THEY CORRECT
RETURN
DISPLAY CHOICE ERROR MESSAGE
DISPLAY ERROR MESSAGE & RANGE
ASK FOR VARIABLE DEVIATIONS

AGEN

ARRAY STRUCTURE

NETWORK: POWER SUPPLY (N=3)

CHARACTERISTIC: REGULATION (C=1)

ARRAY NAME: CARY

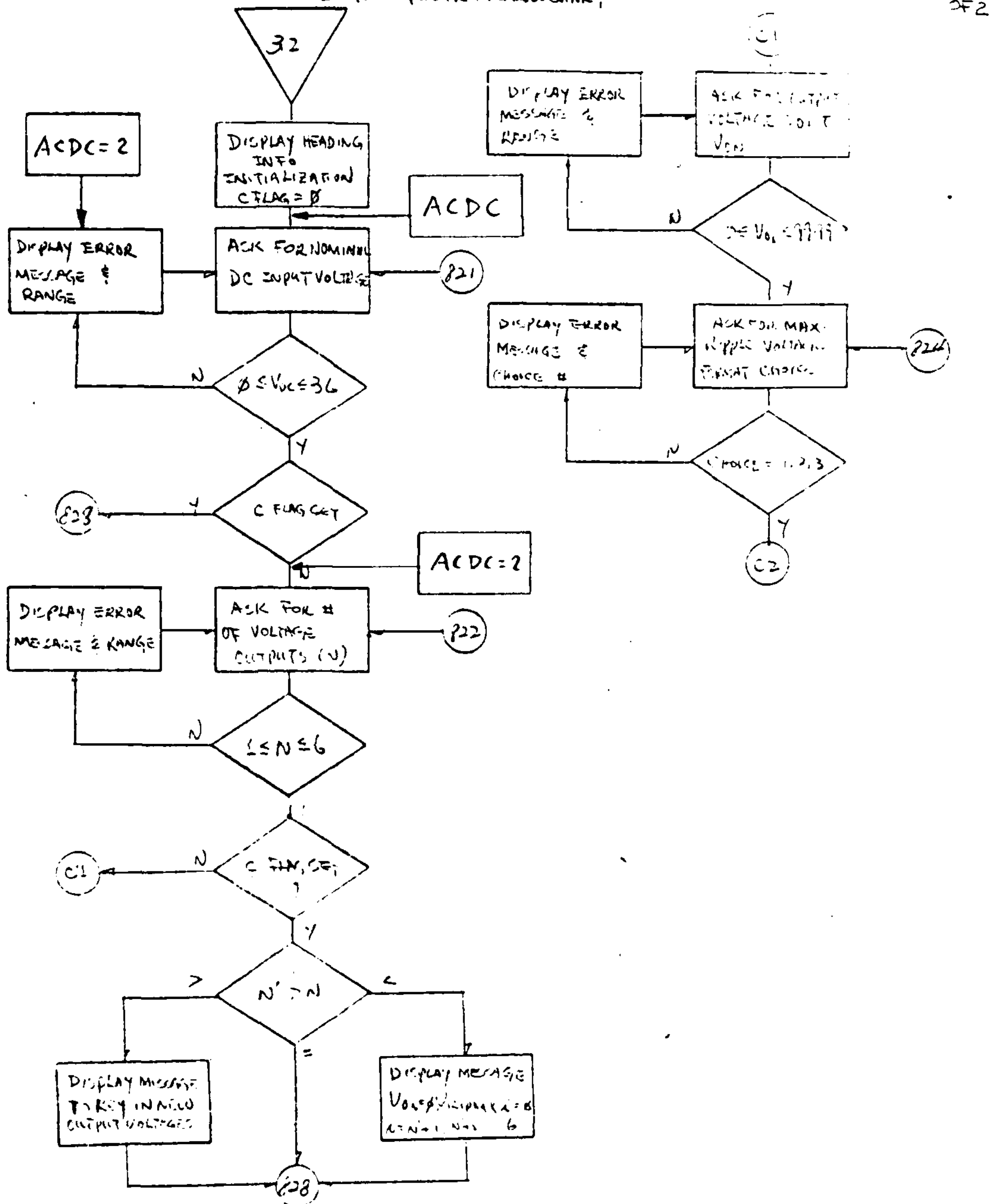
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT31 MODULE.

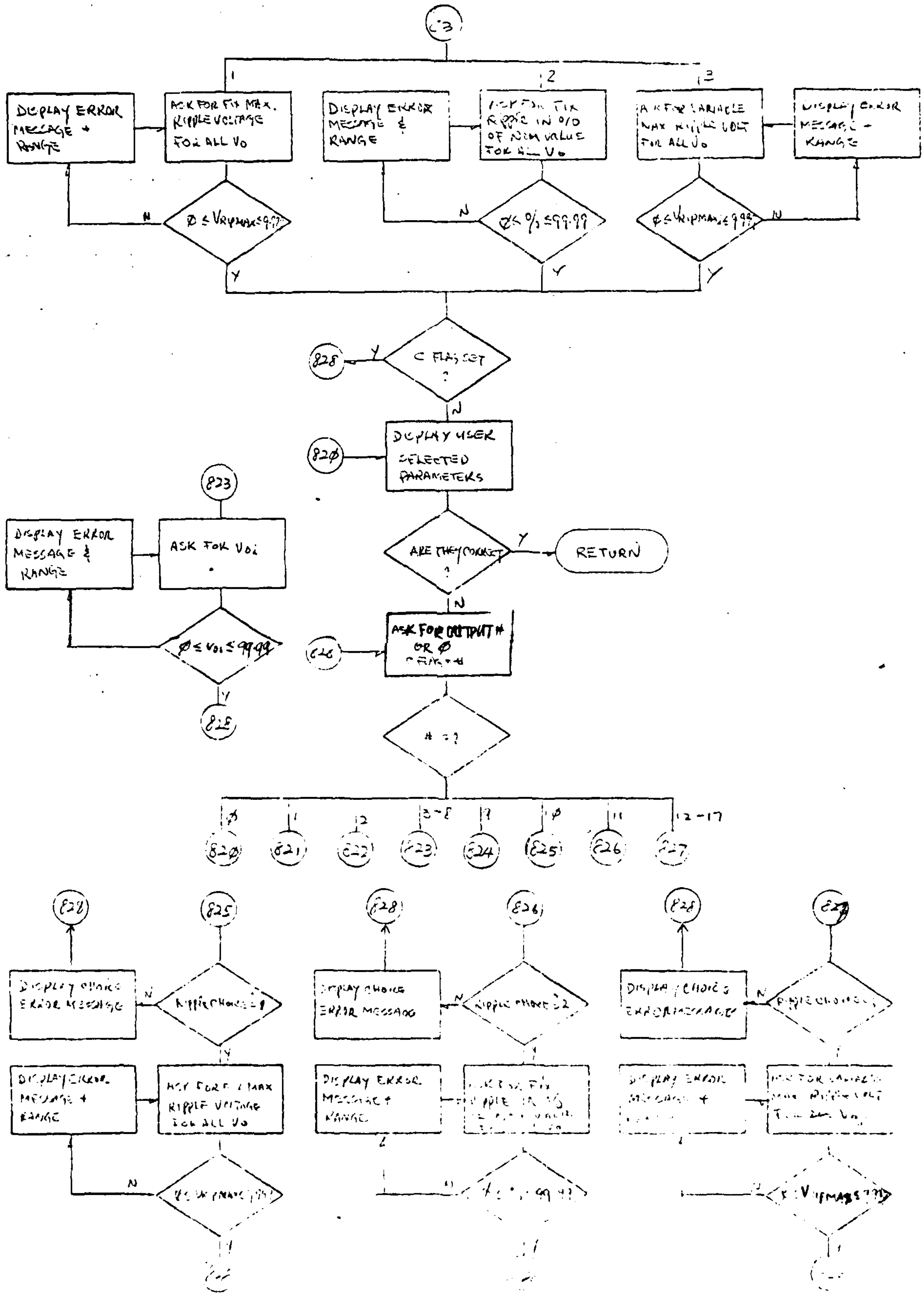
(N)	VARIABLE	EXPLANATION
1	VDC	DC INPUT VOLTAGE.
2	VDCUL	VDC UPPER LIMIT.
3	VDCLL	VDC LOWER LIMIT.
4	N	# OF VDC INPUTS.
—	VOUT(6)	OUTPUT VOLTAGES.
—	OVDF	OUTPUT VOLTAGE DEV FORMAT.
—	FIXDEV	FIXED DEVIATION AMOUNT.
—	VUPDEV(6)	UPPER DEV AMOUNTS.
—	VLODEV(6)	LOWER DEV AMOUNTS.
—	FIXDP	FIXED DEV. %
5 - 10	VOUTUP(6)	OUTPUT VOLTAGE UPPER LIMIT.
11 - 16	VOUTLO(6)	OUTPUT VOLTAGE LOWER LIMIT.
17	ACDC	SIGNAL TYPE 1 or 2 1) Single Phase 60 Hz 115 AC 2) 0-36V DC

IVT3 - ENTRY 32

POWER SUPPLY - RIPPLE MEASUREMENT

DETAIL PROGRAM FLOWCHART




DISPLAY ERROR MESSAGE + RANGE
ASK FOR FIX MAX. RIPPLE VOLTAGE FOR ALL Vo
DISPLAY ERROR MESSAGE & RANGE
ASK FOR FIX RIPPLE IN % OF NOM VALUE FOR ALL Vo
ASK FOR VARIABLE MAX RIPPLE VOLT FOR ALL Vo
C FLAG SET ?
DISPLAY USER SELECTED PARAMETERS
ARE THEY CORRECT ?
RETURN
ASK FOR Voi
DISPLAY ERROR MESSAGE & RANGE
ASK FOR OUTPUT # OR Ø
= ?
DISPLAY CHOICE ERROR MESSAGE
DISPLAY ERROR MESSAGE + RANGE
ASK FOR FIX MAX RIPPLE VOLTAGE FOR ALL Vo

AGEN

ARRAY STRUCTURE

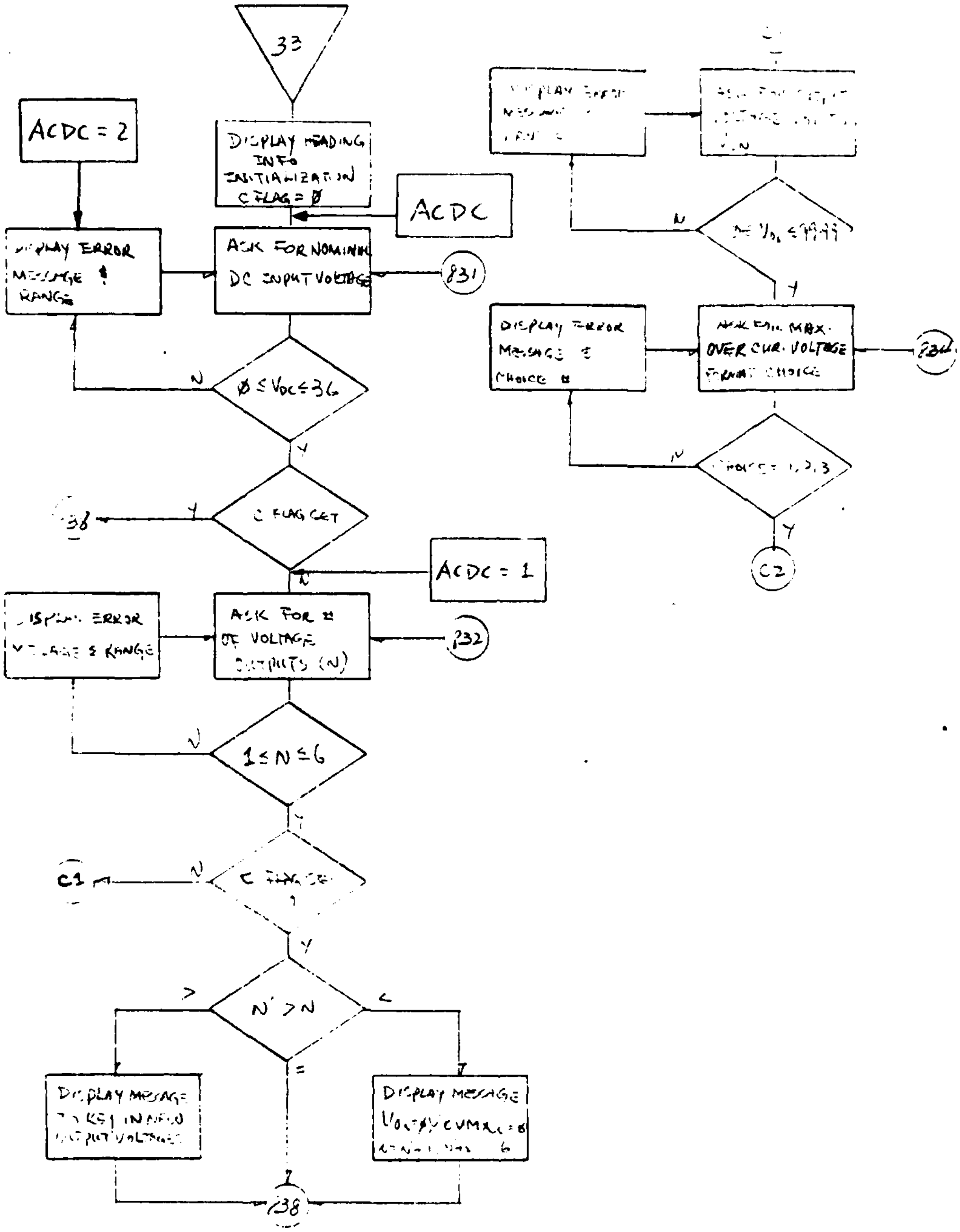
NETWORK: POWER SUPPLY (N=3)

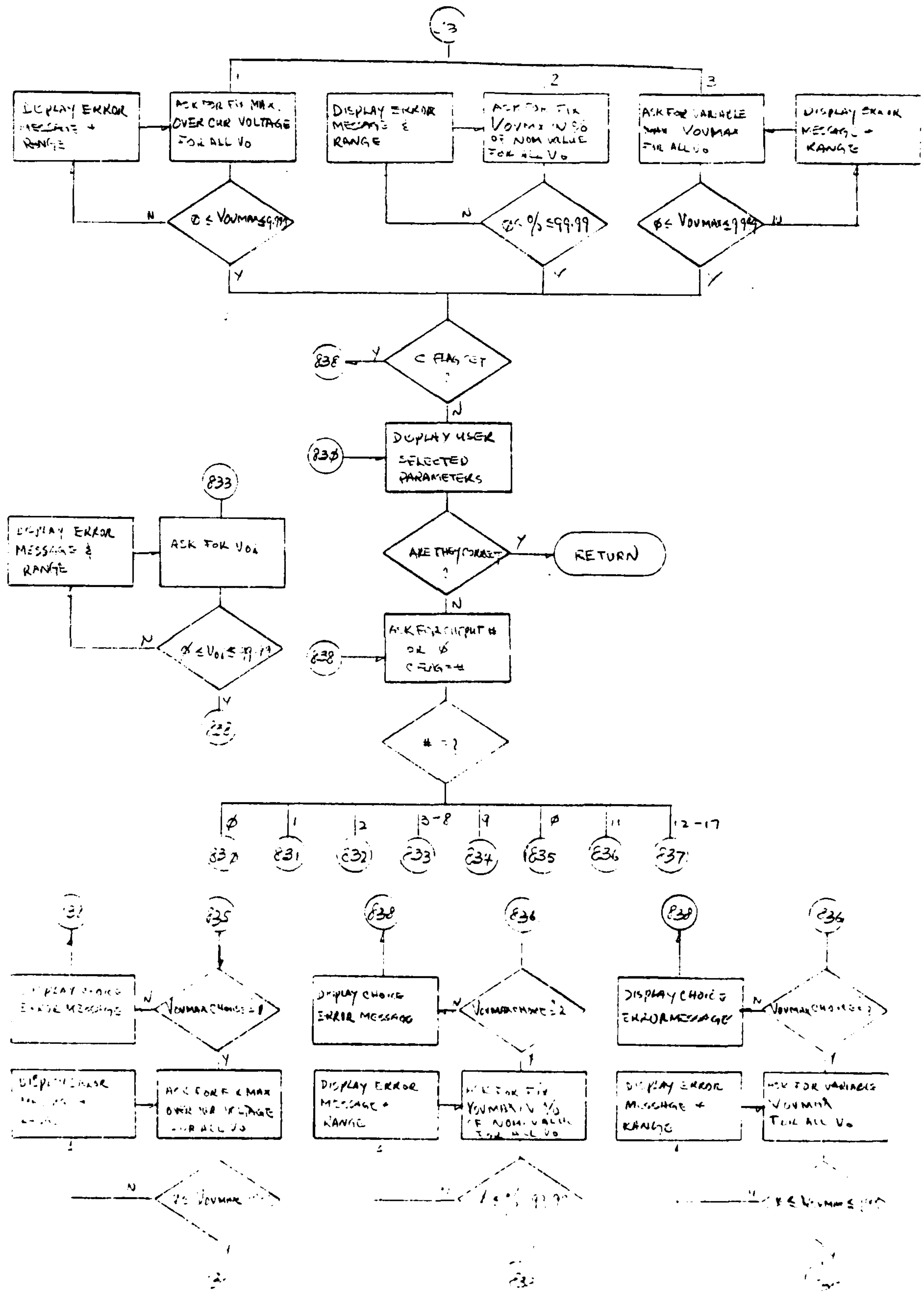
CHARACTERISTIC: RIPPLE MEASUREMENT (C=2)

ARRAY NAME: CARY

PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT32 MODULE.

(N)	VARIABLE	EXPLANATION
1	VDC	NOMINAL DC INPUT VOLTAGE.
2-7	VOUT(6)	OUTPUT VOLTAGES,
8	MRVFC	MAX RIPPLE VOLT. FORMAT CHOICE.
9	FIXMXK	FIXED MAX RIPPLE VOLTAGE AMOUNT.
10	FIXMXP	" " " " %
11-16	VRIPMX(6)	VARIABLE MAX " " AMOUNTS.
17	N	# OF OUTPUT VOLTAGES
18	ACDC	SIGNAL TYPE 1 or 2

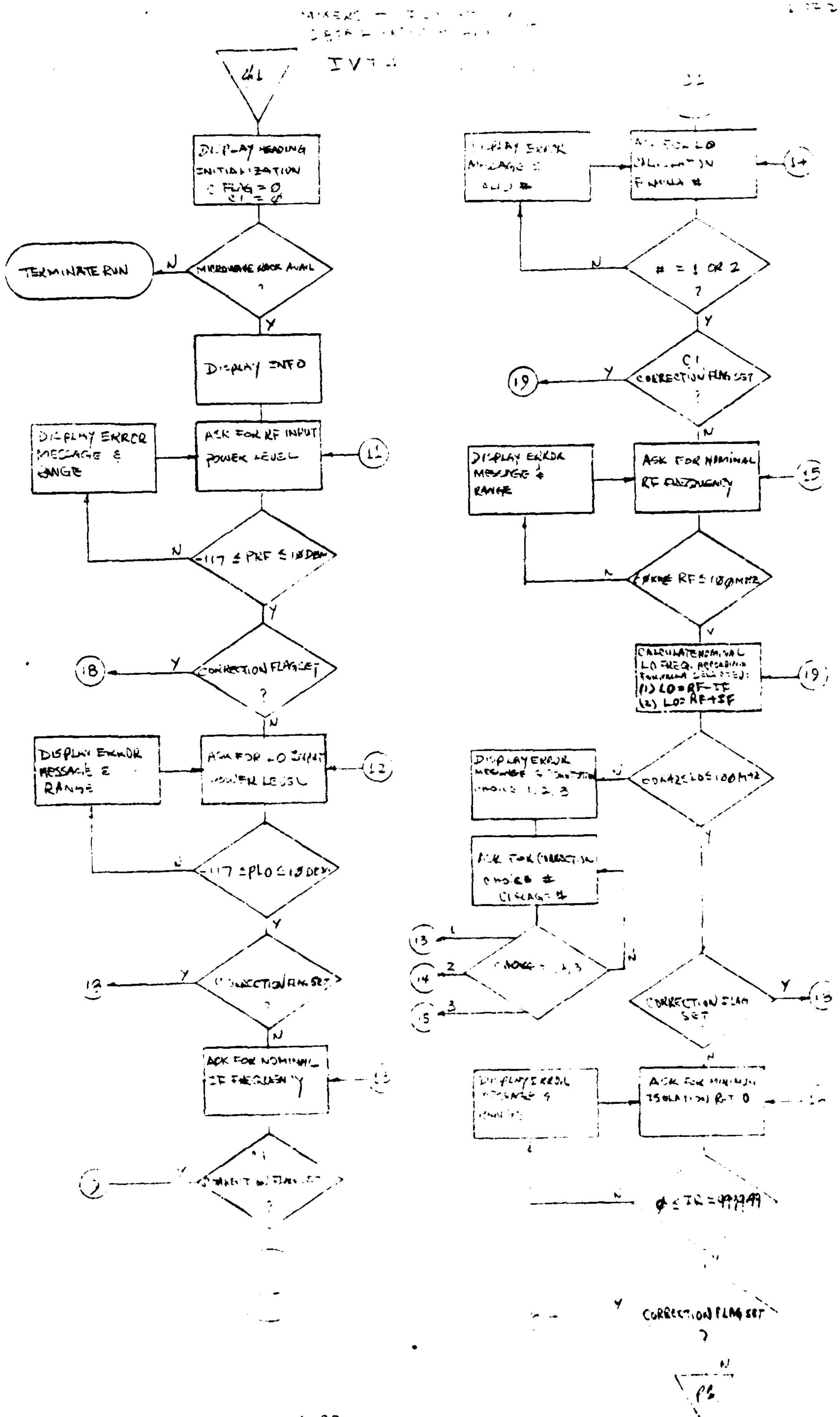

33
ACDC = 2
DISPLAY HEADING INFO INITIALIZATION C FLAG = Ø
ACDC
ASK FOR NOMINAL DC INPUT VOLTAGE
831
Ø ≤ VDC ≤ 36
C FLAG SET
38
ACDC = 1
ASK FOR # OF VOLTAGE OUTPUTS (N)
832
1 ≤ N ≤ 6
C1
N' > N
238
C2
DISPLAY ERROR MESSAGE & CHOICE #
OVER CUR. VOLTAGE FORMAT CHOICE
834


DISPLAY ERROR MESSAGE + RANGE
ASK FOR FIX MAX. OVER CUR VOLTAGE FOR ALL Vo
DISPLAY ERROR MESSAGE & RANGE
DISPLAY ERROR MESSAGE + RANGE
RETURN
DISPLAY USER SELECTED PARAMETERS
ARE THEY CORRECT ?
ASK FOR Vo
DISPLAY CHOICE ERROR MESSAGE
ASK FOR VARIABLE Vovmax FOR ALL Vo

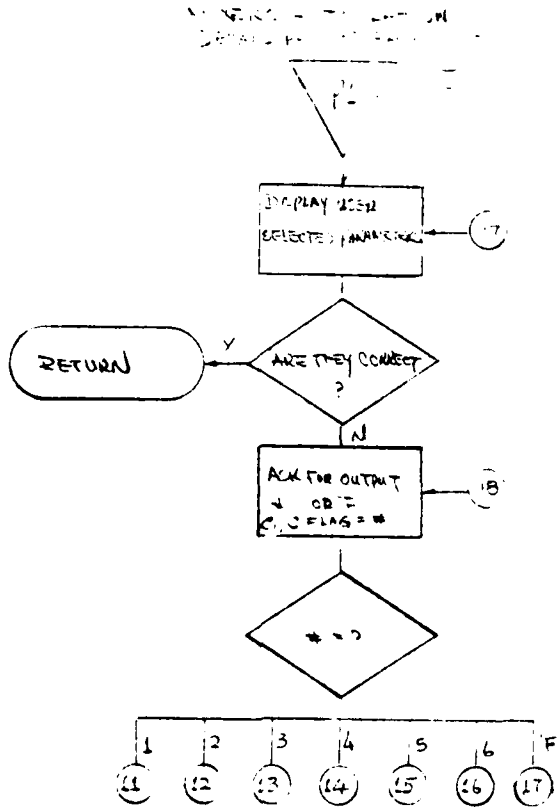
AGEN
ARRAY STRUCTURE

NETWORK: POWER SUPPLY (N=3)
CHARACTERISTIC: OVER CURRENT (C=3)
ARRAY NAME: CARY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT33 MODULE

(N)	VARIABLE	EXPLANATION
1	1 VDC	NOMINAL DC INPUT VOLTAGE.
2	1 V	# OF OUTPUT VOLTAGES.
3-8	6 VOUT(6)	POWER SUPPLY VOLTAGE OUTPUTS.
9-14	6 VMAX(6)	VARIADLE MAXIMUM OVER CURRENT VOLTAGE AMOUNT
10	3 FMT	MAX OVER CURRENT VOLTAGE FORMAT.
11	1 FIXMAX	FIXED MAXIMUM OVER CURRENT VOLTAGE.
12	1 FIXPER	MAX OVER CURRENT VOLTAGE IN %.
15.	ACDC	SIGNAL type 1 or 2 1) AC 2) DC



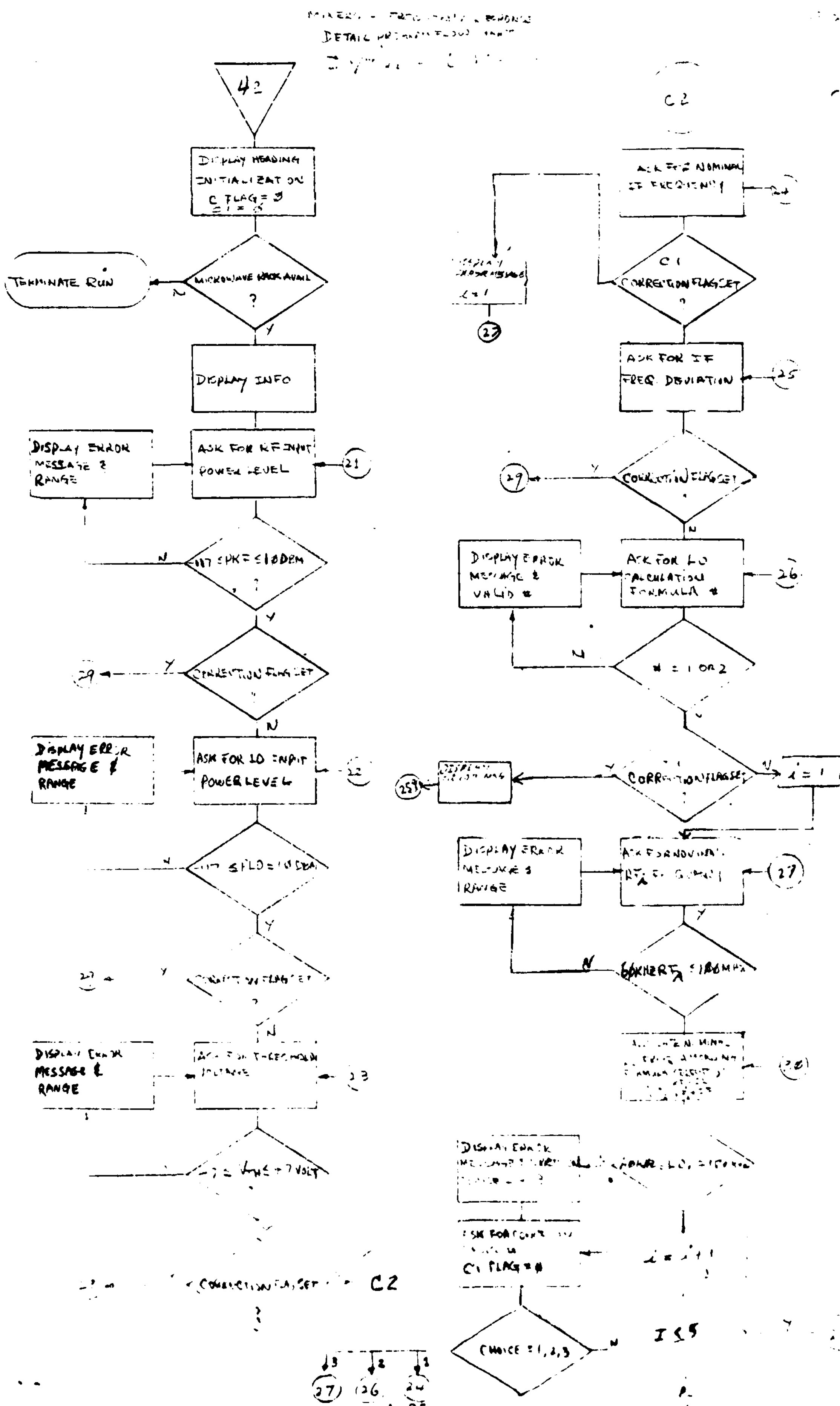
DISPLAY HEADING
INITIALIZATION
TERMINATE RUN
MICROWAVE RACK AVAIL
?
DISPLAY INFO
DISPLAY ERROR MESSAGE & RANGE
ASK FOR RF INPUT POWER LEVEL
CORRECTION FLAG SET
?
ASK FOR LO INPUT POWER LEVEL
ASK FOR NOMINAL IF FREQUENCY
ASK FOR NOMINAL RF FREQUENCY
CALCULATE NOMINAL LO FREQ.
(1) LO=RF-IF
(2) LO=RF+IF
ASK FOR CORRECTION CHOICE #
CORRECTION FLAG SET
CORRECTION FLAG SET
?

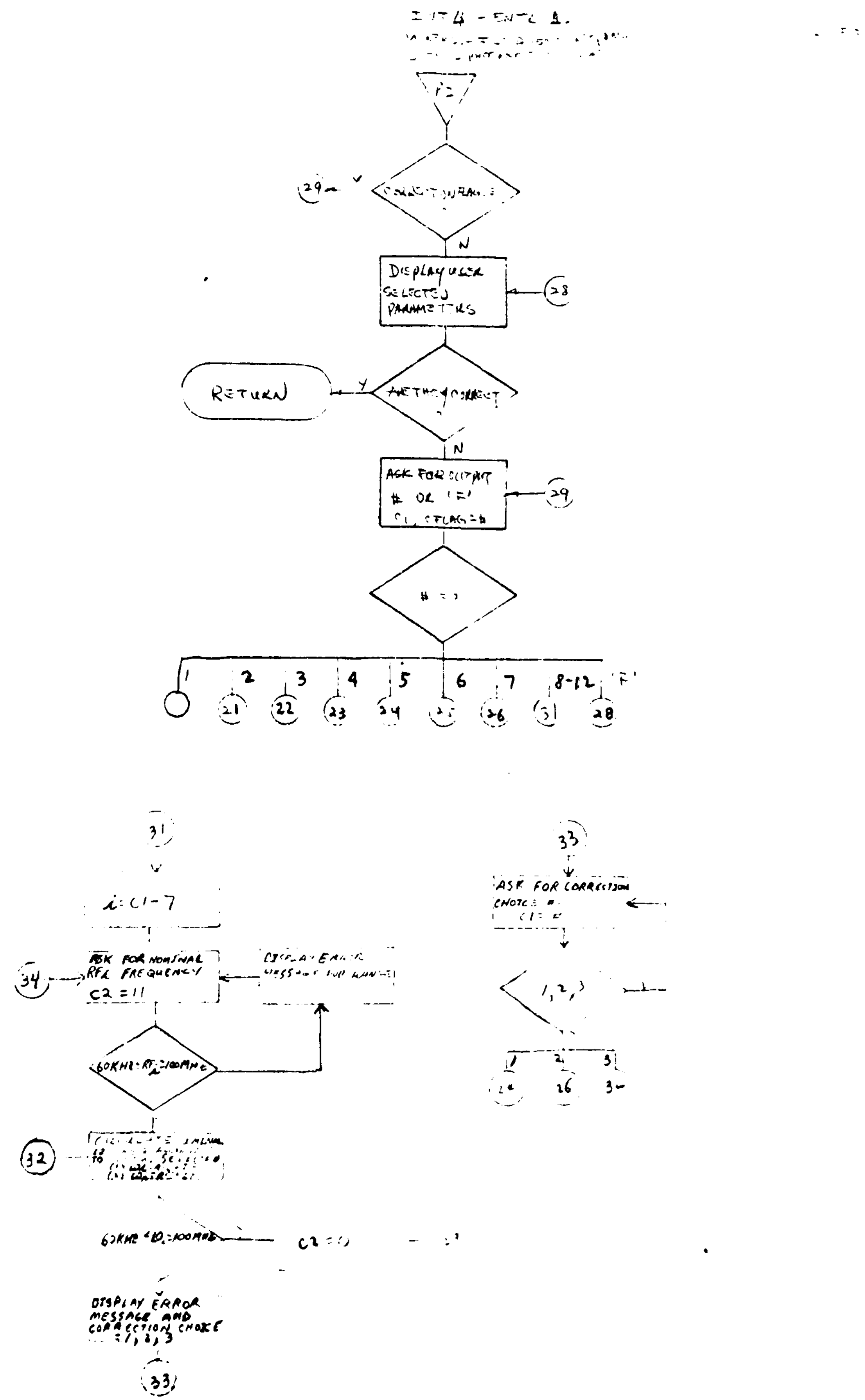

RETURN
ARE THEY CORRECT ?
Y
N
ASK FOR OUTPUT
1
2
3
4
5
6
F
11
12
13
14
15
16
17

AGEN
ARRAY STRUCTURE

NETWORK: MIXER (N=4)
CHARACTERISTIC: ISOLATION (C=1)
ARRAY NAME: CARY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT41 MODULE

(N)	VARIABLE	EXPLANATION
CARY(1)	IR	MINIMUM ISOLATION RATIO
CARY(2)	LO	CALCULATED NOMINAL LO FREQUENCY.
CARY(3)	NIF	NOMINAL IF FREQUENCY.
CARY(4)	PLO	LO INPUT SIGNAL POWER LEVEL.
CARY(5)	PRF	RF INPUT SIGNAL POWER LEVEL,
CARY(6)	RF	NOMINAL RF FREQUENCY.
CARY(7)	DCV	DC VOLTAGE
8	MATLO	INsertion Loss INPUT matching Network RFA
9	MATRF	INsertion Loss INPUT matching network RFB
10	MATOUT	INsertion Loss INPUT matching Network from UUT


42
DISPLAY HEADING
INITIALIZATION
C FLAG = 0
TERMINATE RUN
MICROWAVE HACK AVAIL ?
DISPLAY INFO
DISPLAY ERROR MESSAGE & RANGE
ASK FOR RF INPUT POWER LEVEL
21
CORRECTION FLAG SET ?
29
DISPLAY ERROR MESSAGE & RANGE
ASK FOR LO INPUT POWER LEVEL
22
DISPLAY ERROR MESSAGE & RANGE
23
CORRECTION FLAG SET
C2
CHOICE = 1, 2, 3
27
26
24
C2
ASK FOR NOMINAL IF FREQUENCY
C1 CORRECTION FLAG SET ?
ASK FOR IF FREQ. DEVIATION
25
CORRECTION FLAG SET
29
DISPLAY ERROR MESSAGE & VALID #
ASK FOR LO CALCULATION FORMULA #
26
= 1 OR 2
CORRECTION FLAG SET
i = 1
DISPLAY ERROR MESSAGE & RANGE
27
DISPLAY ERROR MESSAGE & RANGE
i = i + 1
C1 FLAG = 0
I ≤ 5

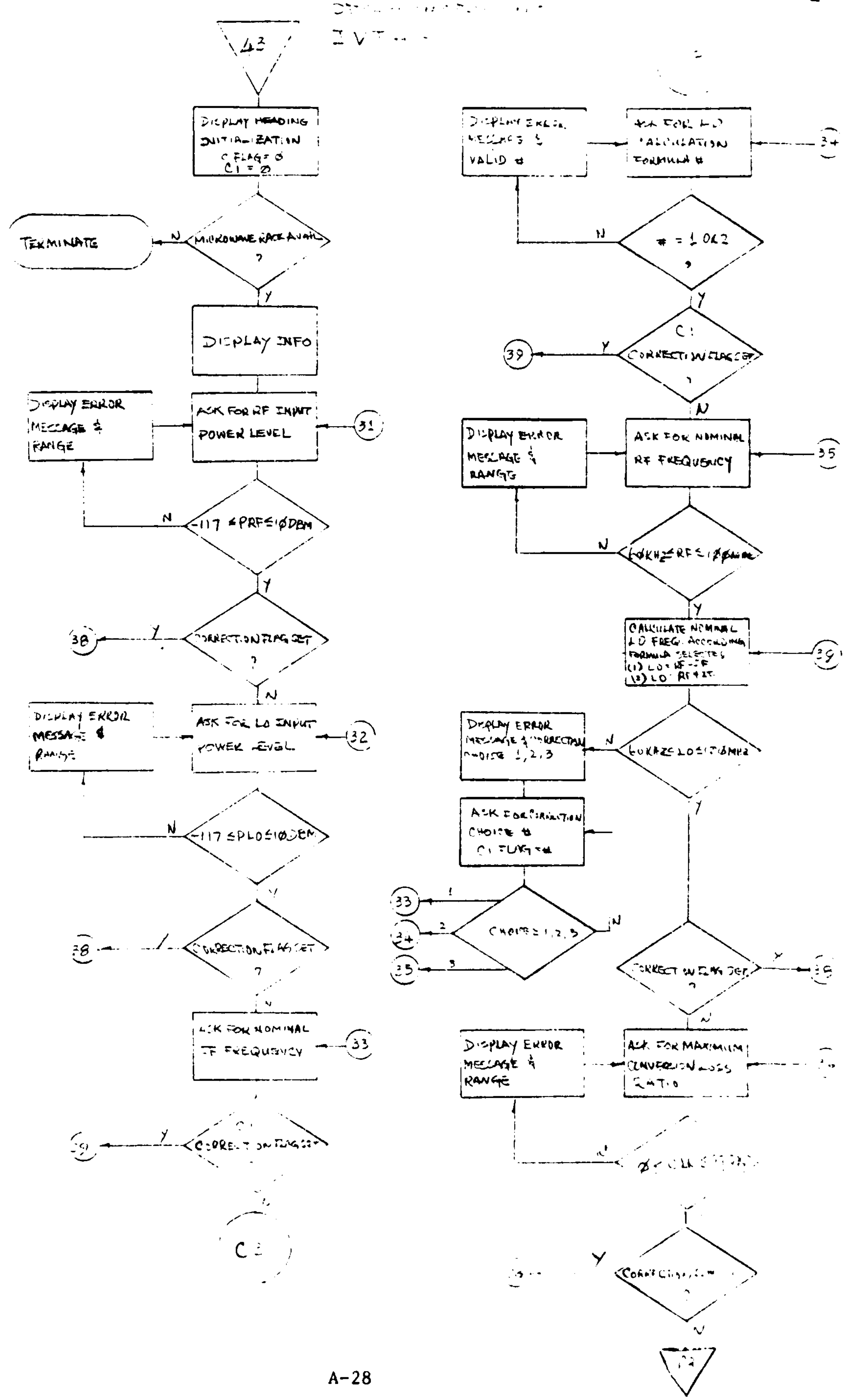


RETURN
DISPLAY USER SELECTED PARAMETERS
ASK FOR OUTPUT # OR 'F'
ASK FOR NOMINAL RFL FREQUENCY
C2 = 11
DISPLAY ERROR MESSAGE AND CORRECTION CHOICE
ASK FOR CORRECTION CHOICE #

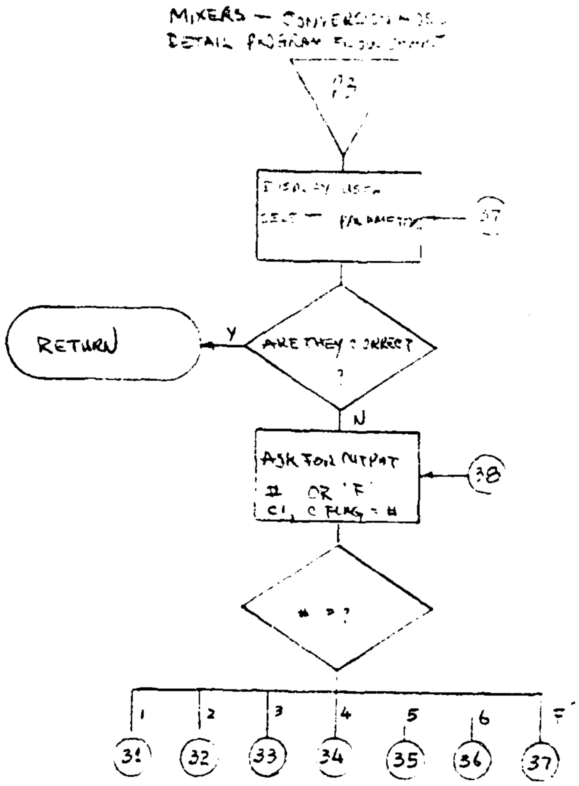
AGEN
ARRAY STRUCTURE

NETWORK: MIXER (N=4)
CHARACTERISTIC: FREQUENCY RESPONSE (C=2)
ARRAY NAME: CARY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OUTPUT MODULE

(N)	VARIABLE	EXPLANATION
1	IFFD	IF FREQUENCY DEVIATION.
2	NOMIF	NOMINAL IF FREQUENCY.
3-7	NOMLO(S)	NOMINAL LO FREQUENCY ELEMENTS.
8-12	NOMRF(S)	NOMINAL RF FREQUENCY.
13	PLO	LO INPUT POWER LEVEL.
14	PRF	RF INPUT POWER LEVEL.
15	VTH	THRESHOLD VOLTAGE.
16	—	IF FREQ. UPPER LIMIT.
17	—	IF FREQ LOWER LIMIT.
18	DCV	DC VOLTAGE.



DISPLAY HEADING INITIALIZATION C FLAG = Ø C1 = Ø
TERMINATE
MICROWAVE RACK AVAIL ?
DISPLAY INFO
DISPLAY ERROR MESSAGE & RANGE
ASK FOR RF INPUT POWER LEVEL
-117 ≤ PRF ≤ 1ØDBM
CORRECTION FLAG SET ?
DISPLAY ERROR MESSAGE & RANGE
ASK FOR LO INPUT POWER LEVEL
-117 ≤ PLO ≤ 1ØDBM
CORRECTION FLAG SET ?
ASK FOR NOMINAL IF FREQUENCY
DISPLAY ERROR MESSAGE & VALID #
ASK FOR LO CALCULATION FORMULA #
= 1 OR 2 ?
CORRECTION FLAG SET ?
DISPLAY ERROR MESSAGE & RANGE
ASK FOR NOMINAL RF FREQUENCY
CALCULATE NOMINAL LO FREQ. ACCORDING FORMULA SELECTED (1) LO = RF - IF (2) LO = RF + IF
ASK FOR CORRECTION CHOICE # C1 FLAG
CHOICE = 1, 2, 3
ASK FOR MAXIMUM CONVERSION LOSS RATIO
DISPLAY ERROR MESSAGE & RANGE



MIXERS —
DETAIL PROGRAM
RETURN
Y
ARE THEY CORRECT
?
N
ASK FOR OUTPUT
OR 'F'
1
2
3
4
5
6
F
31
32
33
34
35
36
37
38

AGEN
ARRAY STRUCTURE

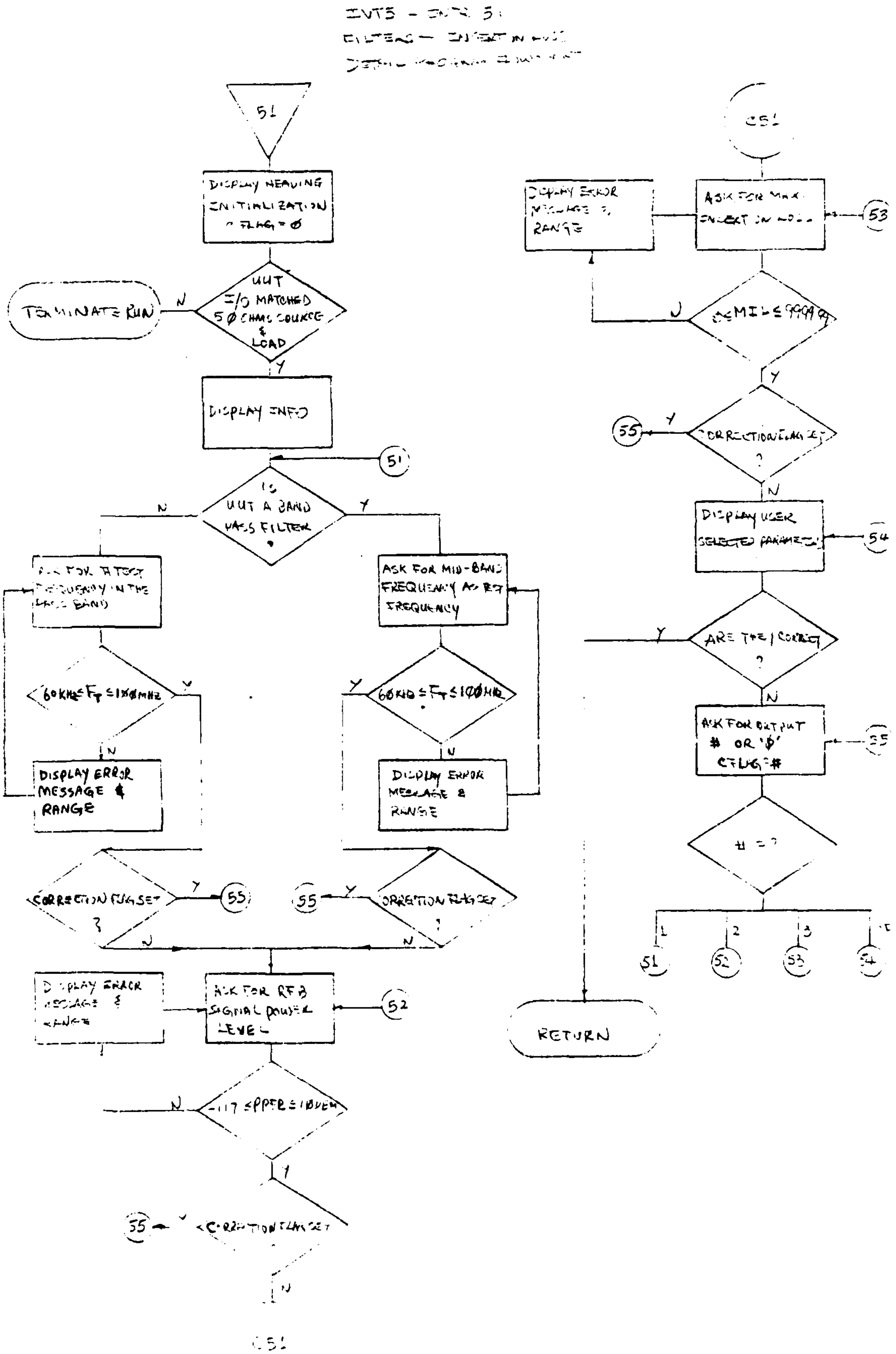
NETWORK: MIXER (N=4)

CHARACTERISTIC: CONVERSION LOSS (C=3)

ARRAY NAME: CARY

PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO CTPT43 MODULE

(N)	VARIABLE	EXPLANATION
1	CLR	MAX. CONVERSION LOSS RATIO.
2	IF	NOMINAL IF FREQUENCY.
3	LO	CALCULATED NOMINAL LO FREQUENCY.
4	PLO	LO INPUT SIGNAL POWER LEVEL.
5	PRF	RF INPUT SIGNAL POWER LEVEL.
6	RF	NOMINAL RF FREQUENCY.
7	DCV	DC VOLTAGE.
16	MATRF	Insertion Loss Input Matching network RFB
17	MATLO	Insertion Loss Input Matching Network RFA
18	MATOUT	Insertion Loss Input matching network from UUT.


51
DISPLAY HEADING INITIALIZATION
UUT I/O MATCHED 50 OHMS SOURCE & LOAD
TERMINATE RUN
DISPLAY INFO
IS UUT A BAND PASS FILTER
ASK FOR MID-BAND FREQUENCY AS REF FREQUENCY
DISPLAY ERROR MESSAGE & RANGE
CORRECTION FLAG SET ?
55
ASK FOR RF SIGNAL POWER LEVEL
52
C51
DISPLAY ERROR MESSAGE & RANGE
53
CORRECTION FLAG SET ?
DISPLAY USER SELECTED PARAMETERS
54
ARE THEY CORRECT ?
ASK FOR OUTPUT # OR '0'
= ?
RETURN

AGEN

ARRAY STRUCTURE

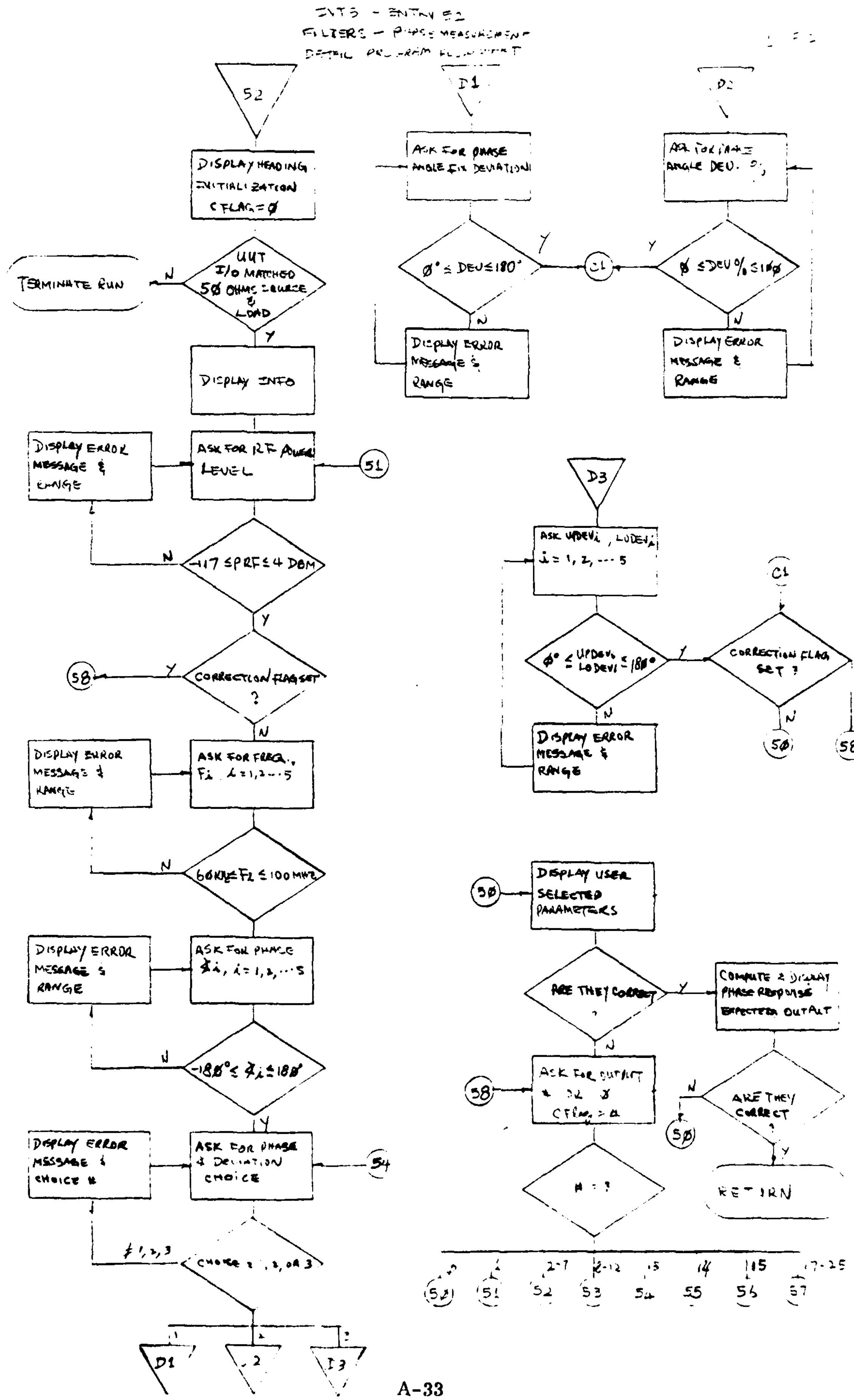
NETWORK: FILTERS (N=5)

CHARACTERISTIC: INSERTION LOSS (C=1)

ARRAY NAME:

PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT51 MODULE.

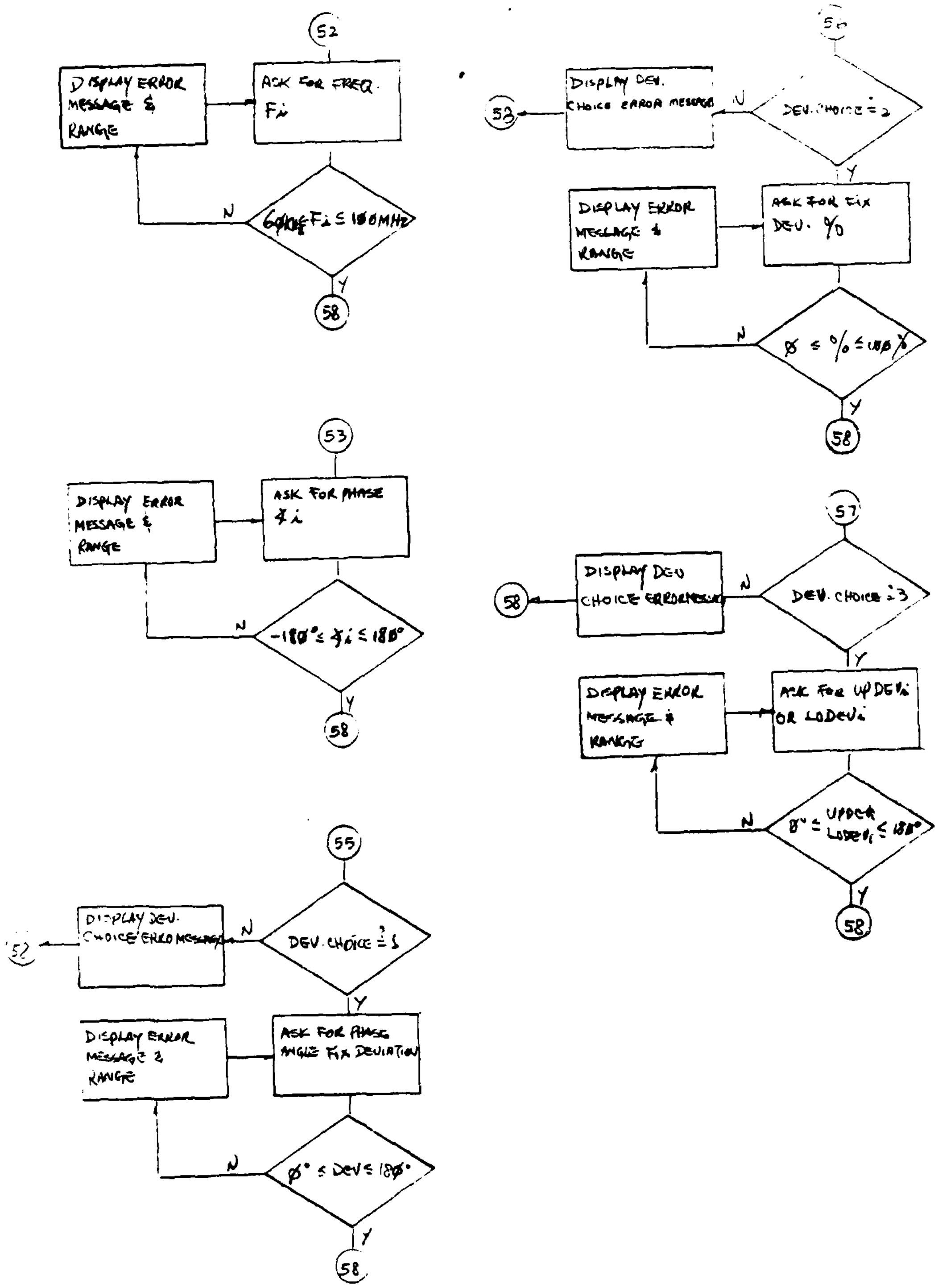
(N)	VARIABLE	EXPLANATION
CARY(1)	TFREQ	TEST FREQUENCY,
CARY(2)	PREB	REB SIGNAL POWER LEVEL.
CARY(3)	MIL	MAXIMUM INSERTION LOSS.
CARY(4)	DCU	DC VOLTAGE TO UUT
5	FCHOIC	Frequency choice
6	SIGLEV	SIGNAL LEVEL
7	MATIN	INPUT MATCHING Network
8	MATOUT	OUTPUT MATCHING Network


52
DISPLAY HEADING
INITIALIZATION
CFLAG=Ø
UUT I/O MATCHED 5Ø OHMS SOURCE & LOAD
TERMINATE RUN
DISPLAY INFO
DISPLAY ERROR MESSAGE & RANGE
ASK FOR RF POWER LEVEL
51
-117 ≤ PRF ≤ 4 DBM
CORRECTION FLAG SET ?
58
DISPLAY ERROR MESSAGE & RANGE
ASK FOR FREQ., Fi, i=1,2,...5
60KHz ≤ Fi ≤ 100MHz
DISPLAY ERROR MESSAGE & RANGE
ASK FOR PHASE Øi, i=1,2,...5
-18Ø° ≤ Øi ≤ 18Ø°
DISPLAY ERROR MESSAGE & CHOICE #
ASK FOR PHASE & DEVIATION CHOICE
54
CHOICE 1, 2, OR 3
D1
D2
D3
D1
ASK FOR PHASE ANGLE FIX DEVIATION
Ø° ≤ DEV ≤ 180°
DISPLAY ERROR MESSAGE & RANGE
C1
D2
ASK FOR PHASE ANGLE DEV. %
Ø ≤ DEV % ≤ 1ØØ
DISPLAY ERROR MESSAGE & RANGE
D3
ASK UPDEVi, LODEVi
i = 1, 2, ... 5
Ø° ≤ UPDEVi, LODEVi ≤ 18Ø°
DISPLAY ERROR MESSAGE & RANGE
CORRECTION FLAG SET ?
5Ø
58
5Ø
DISPLAY USER SELECTED PARAMETERS
ARE THEY CORRECT ?
COMPUTE & DISPLAY PHASE RESPONSE EXPECTED OUTPUT
ARE THEY CORRECT ?
RETURN
58
ASK FOR OUTPUT
CFlag = 1
5Ø
N = ?
2-7
8-12
13
14
15
52
53
54
55
56
57
Y
N

IVTS – [illegible]

FILTERS – PHASE [illegible]

DETAIL PROGRAM FLOWCHART



AGEN
ARRAY STRUCTURE

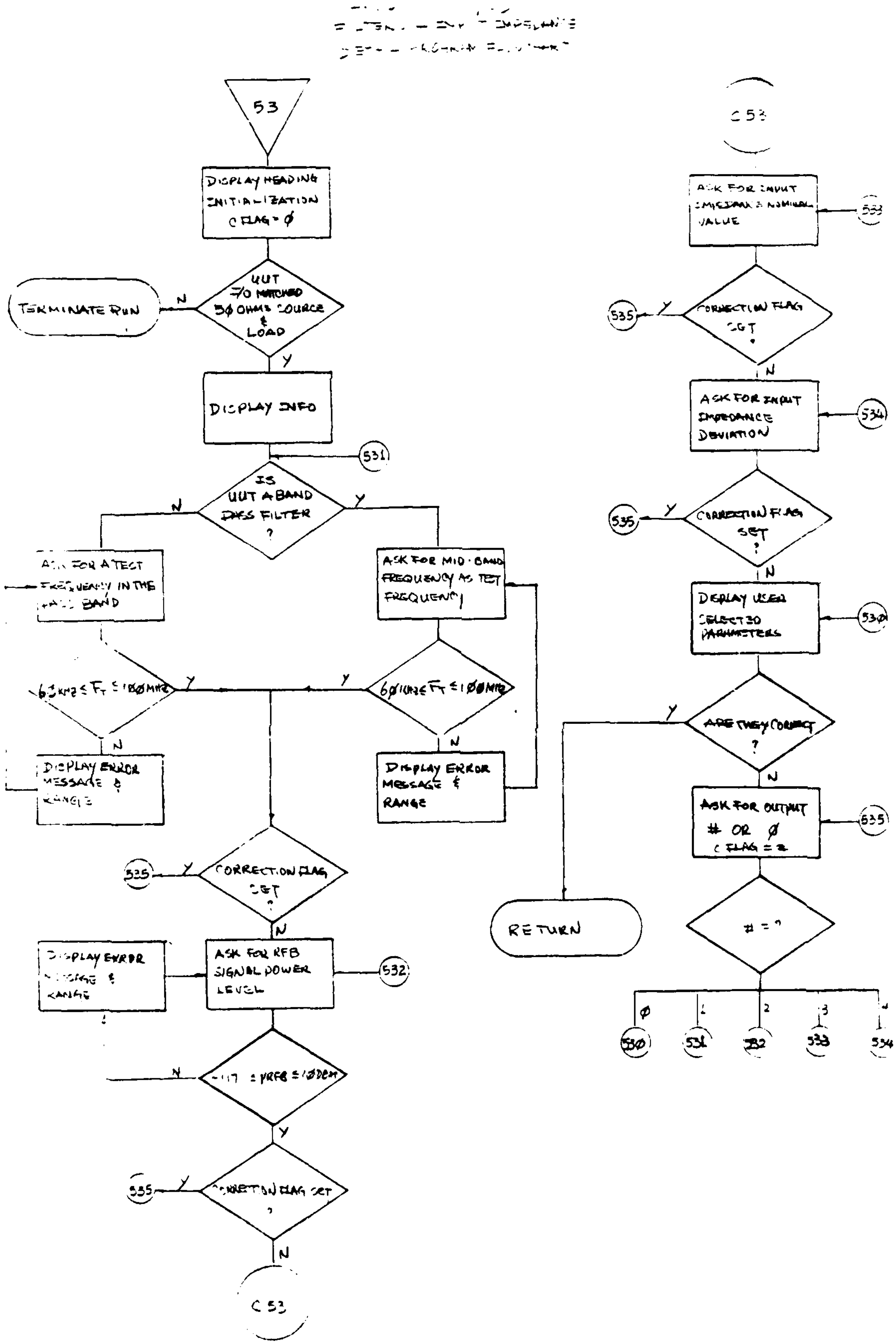
NETWORK: FILTER (I =)

CHARACTERISTIC: PHASE MEASUREMENT (L = 2)

ARRAY NAME: CARY

PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT52 MODULE

(N)	VARIABLE	EXPLANATION
1	FIXDEV	FIXED DEVIATION AMOUNT.
2 - 6 -	FREQ(5)	FREQUENCIES.
7 - 11	LODEV(5)	LOWER DEV. FOR 5 PHASE ANGLES.
12	PADEV	PHASE ANGLE DEVIATION %.
13	PADF	PHASE ANGLE DEVIATION FORMAT.
14 - 18	PANGL1(5)	PHASE ANGLES 1 - 5.
19 - 28	PANGL2(10)	PHASE ANGLE UPPER & LOWER LIMIT PANGL2(1), PANGL2(2) CONTAIN LIMITS FOR 1ST PHASE ANGLE ETC
29	PRF	RF POWER LEVEL.
30 - 34	UPDEV(5)	UPPER DEV. FOR 5 PHASE ANGLES
35	DCV	DC VOLTAGE TO UUT
36	FCHOIC	Frequency Choice
37	SIGLEV	SIGNAL LEVEL

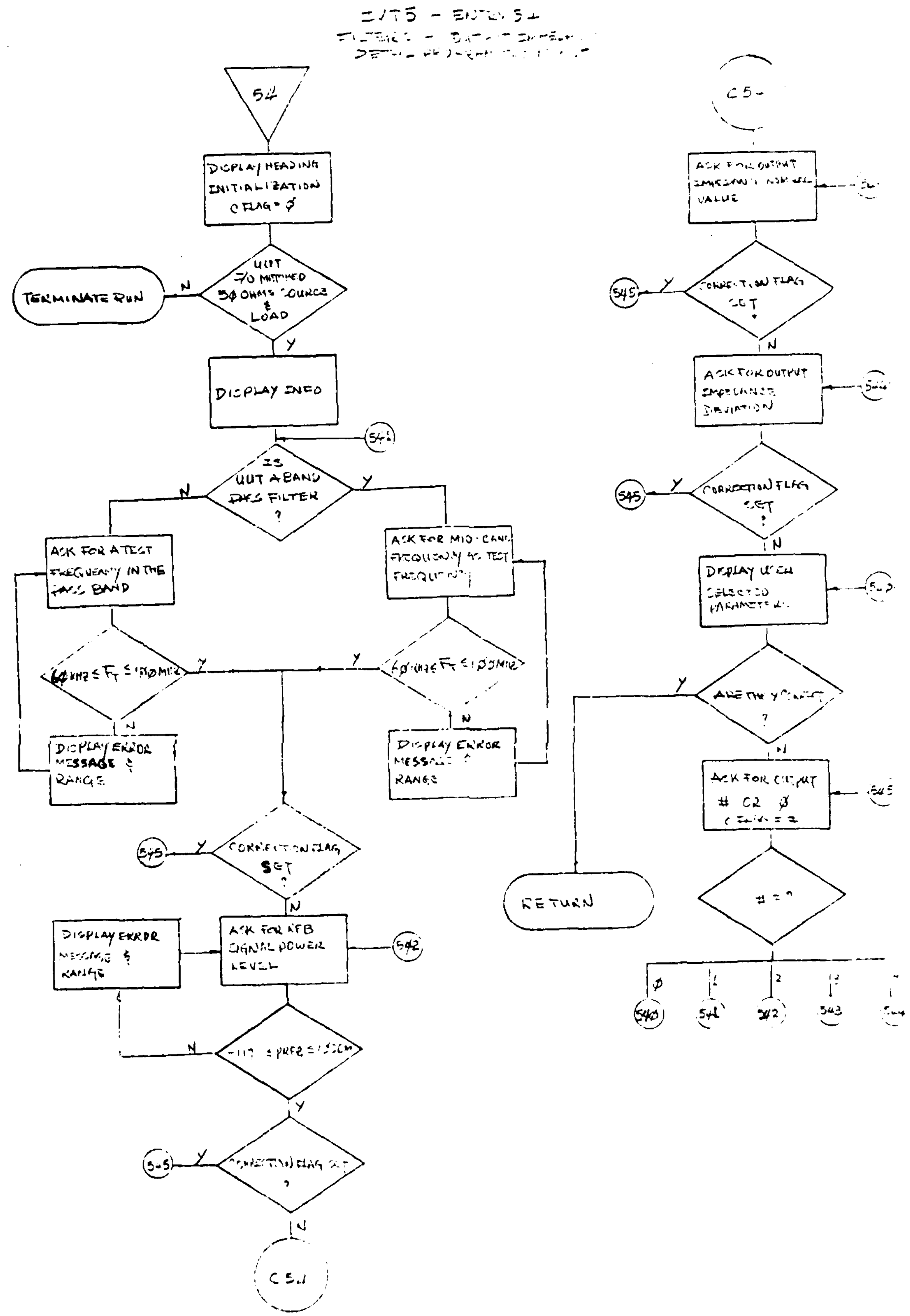


53
DISPLAY HEADING
INITIALIZATION
C FLAG = Ø
UUT
Z/O MATCHED
5Ø OHMS SOURCE
&
LOAD
TERMINATE RUN
N
Y
DISPLAY INFO
531
IS
UUT A BAND
PASS FILTER
?
N
Y
ASK FOR MID-BAND
FREQUENCY AS TEST
FREQUENCY
6Ø KHZ ≤ F_T ≤ 1ØØ MHZ
DISPLAY ERROR
MESSAGE &
RANGE
DISPLAY ERROR
MESSAGE &
RANGE
CORRECTION FLAG
SET
?
535
ASK FOR RFB
SIGNAL POWER
LEVEL
532
DISPLAY ERROR
MESSAGE &
RANGE
CORRECTION FLAG SET
?
535
C 53
C 53
ASK FOR INPUT
IMPEDANCE NOMINAL
VALUE
533
CORRECTION FLAG
SET
535
ASK FOR INPUT
IMPEDANCE
DEVIATION
534
CORRECTION FLAG
SET
?
535
DISPLAY USER
SELECTED
PARAMETERS
53Ø
ARE THEY CORRECT
?
RETURN
ASK FOR OUTPUT
OR Ø
535
53Ø
531
532
533
534

MGEN
ARRAY STRUCTURE

NETWORK: FILTERS (N=5)
CHARACTERISTIC: INPUT IMPEDANCE (C=3)
ARRAY NAME: CARY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT53 MODULE

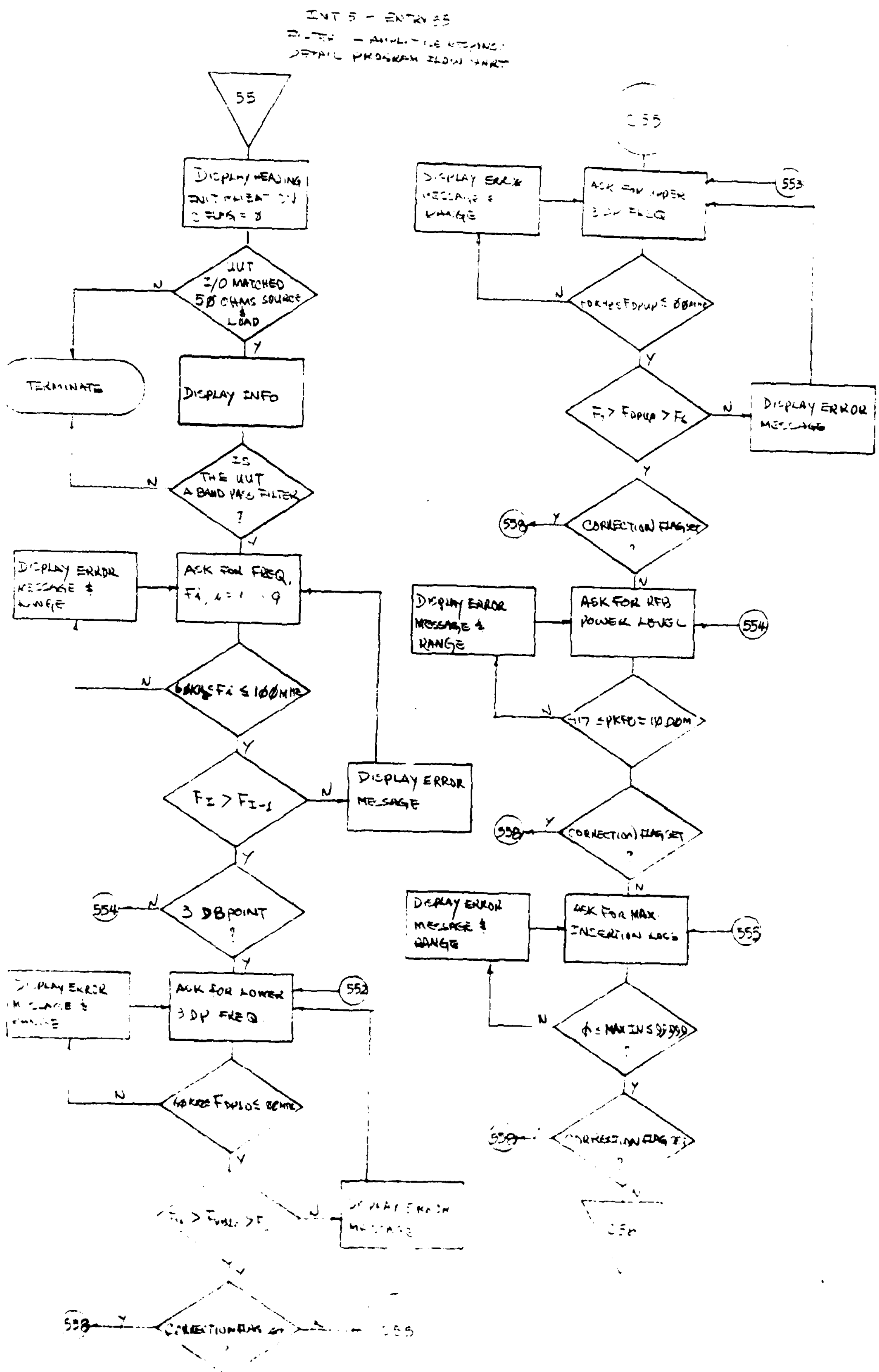
(N)	VARIABLE	EXPLANATION
CARY(1)	IID	INPUT IMPEDANCE DEVIATION.
CARY(2)	IINV	INPUT IMPEDANCE NOMINAL VALUE
CARY(3)	PRFB	REQUIRED SIGNAL POWER LEVEL FROM RFG TO UUT INPUT.
CARY(4)	TFREQ	TEST FREQUENCY.
CARY(5)	—	INPUT IMPEDANCE UPPER LIMIT.
CARY(6)	—	INPUT IMPEDANCE LOWER LIMIT.
CARY(7)	DCV	DC VOLTAGE TO UUT
8	FCHOIC	Frequency Choice
9	SIGLEV	SIGNal Level
10	RESIST	Load Resistance IN IO


54
DISPLAY HEADING
INITIALIZATION
C FLAG = Ø
UUT
I/O MATCHED
5Ø OHMS SOURCE
&
LOAD
TERMINATE RUN
N
Y
DISPLAY INFO
IS
UUT A BAND
PASS FILTER
?
ASK FOR A TEST
FREQUENCY IN THE
PASS BAND
DISPLAY ERROR
MESSAGE &
RANGE
DISPLAY ERROR
MESSAGE &
RANGE
CORRECTION FLAG
SET
?
545
ASK FOR RFB
SIGNAL POWER
LEVEL
542
DISPLAY ERROR
MESSAGE &
RANGE
CORRECTION FLAG SET
?
C54
C5
CORRECTION FLAG
SET
ASK FOR OUTPUT
IMPEDANCE
DEVIATION
CORRECTION FLAG
SET
?
DISPLAY
SELECTED
RETURN
Ø
540
2
542
543

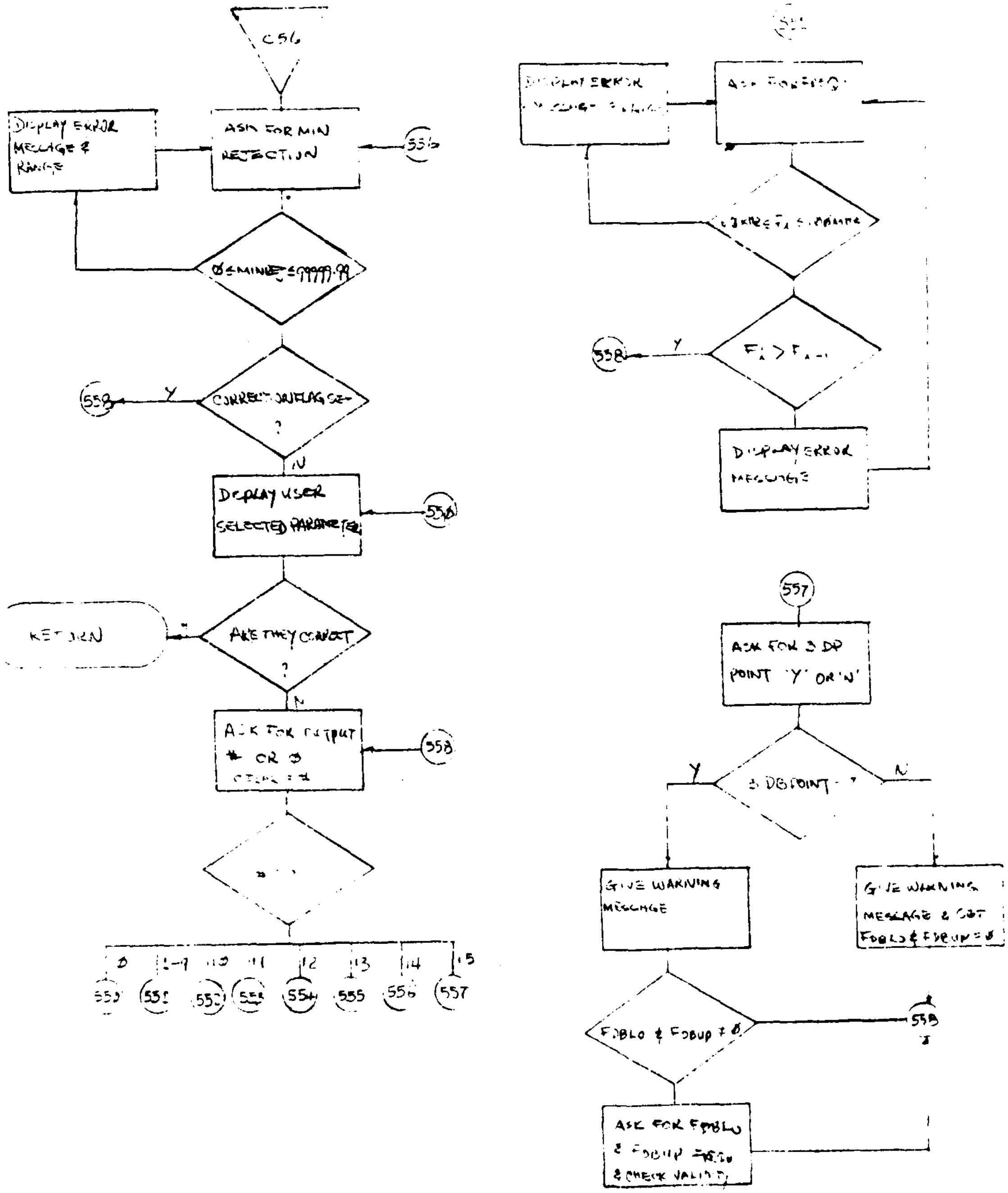
AGEN
ARRAY STRUCTURE

NETWORK: FILTERS (N=5)
CHARACTERISTIC: OUTPUT IMPEDANCE (C=4)
ARRAY NAME: CARY
PURPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTT54 MODULE

(N)	VARIABLE	EXPLANATION
CARY(1)	OID	OUTPUT IMPEDANCE DEVIATION.
CARY(2)	OINV	OUTPUT IMPEDANCE NOM. VALUE
CARY(3)	PRFB	RFB SIGNAL POWER LEVEL.
CARY(4)	TFREQ	TEST FREQUENCY.
CARY(5)	—	UPPER LIMIT OUTPUT IMPEDANCE
CARY(6)	—	LOWER LIMIT OUTPUT IMPEDANCE
CARY(7)	DCV	DC VOLTAGE TO UUT.
8	FCHOIC	Frequency choice
9	OHM	Load IN ID RT
10	Resist	Load IN ID RL
11	SIGLEV	Signal level MVS


55
DISPLAY HEADING
UUT I/O MATCHED 50 OHMS SOURCE & LOAD
TERMINATE
DISPLAY INFO
IS THE UUT A BAND PASS FILTER ?
DISPLAY ERROR MESSAGE & RANGE
ASK FOR FREQ.
DISPLAY ERROR MESSAGE
3 DB POINT ?
554
ASK FOR LOWER 3 DB FREQ
553
DISPLAY ERROR MESSAGE
558
CORRECTION FLAG SET ?
DISPLAY ERROR MESSAGE & RANGE
ASK FOR RFB POWER LEVEL
CORRECTION FLAG SET ?
DISPLAY ERROR MESSAGE & RANGE
ASK FOR MAX INSERTION LOSS
555

INT5 - ENTRY 55



AGEN
ARRAY STRUCTURE

FILTER

ETWORK: (N = 5)

ARACTERISTIC: AMPLITUDE RESPONSE (C = 5)

RRAY NAME: CARY

URPOSE: TO STORE PARAMETRIC VALUES IN INTEGER FORM AND PASS THEM TO OTPT55 MODULE

N)	VARIABLE	EXPLANATION
	ARFREQ(9)	FREQUENCIES
	DB3	CONTAIN Y OR N IN RESPONSE TO 3 DB POINT TEST REQUEST.
	FDBLO	LOWER 3DB POINT FREQ.
	FDBUP	UPPER 3DB POINT FREQ.
	PRFB	RFB POWER LEVEL.
	MAXIL	MAX. INSERTION LOSS
	MINREJ	MINIMUM REJECTION.
	DCV	DC VOLTAGE TO UUT.
	FCHOIC	Frequency Choice
	SIGLEV	SIGNal Level
	MATIN	INPUT MATCHING NETWORK
	MATOUT	OUTPUT MATCHING NETWORK

ND

-80